



**TUGAS AKHIR - TE 141599**

**RANCANG BANGUN SISTEM KANTOR PINTAR  
BERBASIS “INTERNET OF THINGS”**

Tri Suliswanto  
NRP 2214 105 019

Dosen Pembimbing  
Dr. Ir. Wirawan, DEA.  
Dr. Istas Pratomo, S.T., M.T.

JURUSAN TEKNIK ELEKTRO  
Fakultas Teknologi Industri  
Institut Teknologi Sepuluh Nopember  
Surabaya 2016



**FINAL PROJECT - TE 141599**

**DESIGN AND IMPLEMENTATION SMART OFFICE  
SYSTEM BASED ON INTERNET OF THINGS**

Tri Suliswanto  
NRP 2214 105 019

Supervisors  
Dr. Ir. Wirawan, DEA.  
Dr. Istas Pratomo, S.T., M.T.

DEPARTMENT OF ELECTRICAL ENGINEERING  
Faculty of Industrial Technology  
Sepuluh Nopember Institute of Technology  
Surabaya 2016

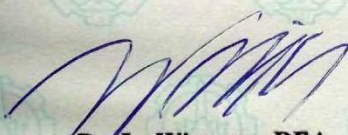
# **RANCANG BANGUN SISTEM KANTOR PINTAR BERBASIS “INTERNET OF THINGS”**

## **TUGAS AKHIR**

**Diajukan Guna Memenuhi Sebagian Persyaratan Untuk  
Memperoleh Gelar Sarjana Teknik Elektro  
Pada  
Bidang Studi Telekomunikasi Multimedia  
Jurusan Teknik Elektro  
Institut Teknologi Sepuluh Nopember**

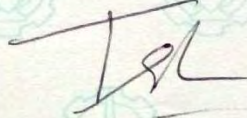
**Menyetujui**

**Dosen Pembimbing I,**



**Dr. Ir. Wirawan, DEA.**  
**NIP. 196311091989031011**

**Dosen Pembimbing II,**



**Dr. Istas Pratomo, S.T., M.T.**  
**NIP. 197903252003121001**



# **RANCANG BANGUN SISTEM KANTOR PINTAR BERBASIS “INTERNET OF THINGS”**

Tri Suliswanto  
2214 105 019

Dosen Pembimbing I : Dr. Ir. Wirawan, DEA.  
Dosen Pembimbing II : Dr. Istas Pratomo, ST., MT.

## **ABSTRAK**

Saat ini kehidupan manusia memiliki tingkat mobilitas yang cukup tinggi, hal ini mendorong terciptanya berbagai inovasi dibidang teknologi untuk memenuhi kebutuhan mobilitas tersebut, salah satunya adalah inovasi teknologi *Internet of Things (IoT)*. Teknologi *Internet of Things* menawarkan kemudahan dan kenyamanan mengakses segala sesuatu dari mana saja, kapan saja dengan bantuan koneksi jaringan internet. Salah satu pengaplikasian *Internet of Things* dapat diterapkan di sebuah gedung (kantor pintar) atau rumah (rumah pintar), teknologi ini memungkinkan kita dapat mengendalikan dan memantau perangkat elektronik dari penjuru dunia.

Tujuan dari Tugas Akhir ini adalah untuk merancang sistem kantor pintar dengan memanfaatkan teknologi *Internet of Things* sebagai solusi kebutuhan manusia akan mobilitas untuk pemantauan dan pengendalian perangkat-perangkat berbasis elektronik.

Hasil perancangan berupa 3 buah node sensor dimana masing-masing node memiliki sensor yang berbeda-beda, diantaranya adalah sensor intensitas cahaya (LDR), sensor temperatur ruangan, *infrared transmitter*, *infrared proximity sensor*, *magnetic door sensor*. Masing-masing sensor terintegrasi dengan modul wifi 802.11 sebagai modul komunikasi antara node dengan node gateway, node gateway mentransmisikan data ke server dan data dari server diakses oleh *mobile client* via telepon pintar.

**Kata Kunci :**IoT, *Smart Home*, *sensor*, *Internet*.

# **DESIGN AND IMPLEMENTATION SMART OFFICE SYSTEM BASED ON INTERNET OF THINGS**

Tri Suliswanto  
2214 105 019

Supervisor I : Dr. Ir. Wirawan, DEA.  
Supervisor II : Dr. Istas Pratomo, ST., MT.

## **ABSTRACT**

Currently human life has a mobility rate is high enough, it encourages the creation of many innovations in technology to meet the mobility needs, one of which is the technological innovation the Internet of Things (IOT). Internet of Things technology offers the ease and convenience of access everything from anywhere, anytime with the help of the Internet network connection. One application of the Internet of Things can be applied in a building (smart office) or home (smart home), this technology allows us to control and monitor electronic devices from around the world.

The purpose of this final project is to design a smart office system by utilizing the Internet of Things as a human need for mobility solutions for the monitoring and control of devices based electronics.

The results of the design in the form of three sensor nodes where each node has a different sensor, such as light intensity sensor (LDR), room temperature sensors, infrared transmitters, infrared proximity sensors, magnetic door sensor. Each sensor module is integrated with wifi 802.11 as communication modules between the nodes by node gateway, the gateway node transmits the data to the server and the data from the server accessed by the client via mobile smart phones.

**Keyword :** IoT, Smart Home, sensor, Internet

## DAFTAR ISI

<b>ABSTRAK .....</b>	<b>i</b>
<b>ABSTRACT .....</b>	<b>iii</b>
<b>KATA PENGANTAR.....</b>	<b>v</b>
<b>DAFTAR ISI.....</b>	<b>vii</b>
<b>DAFTAR GAMBAR.....</b>	<b>xi</b>
<b>DAFTAR TABEL .....</b>	<b>xv</b>
<b>BAB 1 PENDAHULUAN .....</b>	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah.....	1
1.3 Batasan Masalah .....	2
1.4 Tujuan.....	2
1.5 Metodologi.....	2
1.6 Sistematika Pembahasan .....	3
1.7 Relevansi.....	4
<b>BAB 2 IOT SMART OFFICE DAN TEORI PENUNJANG PERANCANGANNYA.....</b>	<b>5</b>
2.1 Konsep Smart Office.....	5
2.2 Internet of Things.....	6
2.3 Aplikasi Server Internet of Things.....	7
2.4 Aplikasi Android.....	8
2.5 Protokol Komunikasi .....	8
2.5.1 Lapisan OSI .....	9
2.5.2 Protokol HTTP .....	9
2.5.3 Protokol UDP .....	11
2.5.3.1 <i>Frame Data UDP</i> .....	11
2.5.3.2 <i>Karakteristik UDP</i> .....	11
2.5.3.3 <i>Layanan antar host yang tidak disediakan UDP</i> ....	12
2.6 Sistem Tertanam ( <i>Embedded System</i> ) .....	12
2.6.1 AVR Mikrokontroller .....	13
2.6.2 <i>Single Board Computer (SBC)</i> .....	15
2.6.3 Modul Wifi .....	16
2.6.4 Sensor .....	17
2.6.4.1 <i>Sensor Temperatur dan Kelembaban</i> .....	17
2.6.4.2 <i>Sensor Intensitas Cahaya</i> .....	18
2.6.4.3 <i>Sensor Infrared Proximity</i> .....	20

2.6.5	Aktuator.....	21
2.6.5.1	<i>Kunci Pintu Elektrik</i> .....	21
2.6.5.2	<i>Driver Lampu</i> .....	22
<b>BAB 3 PERANCANGAN SISTEM KANTOR PINTAR .....</b>		<b>23</b>
3.1	Diagram Alir Perancangan .....	23
3.2	Rancangan Sistem.....	24
3.2.1	Rancangan Node Kontroller Temperatur .....	28
3.2.2	Rancangan Node Kontroller Pintu dan Sensor <i>Visitor Counter</i> .....	28
3.2.3	Rancangan Node Kontroller Pencahayaan Ruangan ....	29
3.2.4	Rancangan Node Gateway .....	29
3.2.5	Rancangan Aplikasi Android .....	30
3.3	Implementasi dan Konfigurasi Node .....	31
3.3.1	Proses Konfigurasi Server IoT .....	31
3.3.1.1	<i>Konfigurasi Channel</i> .....	32
3.3.1.2	<i>Konfigurasi TalkBack</i> .....	33
3.3.2	Implementasi dan Konfigurasi Node Kontroller Temperature .....	35
3.3.3	Implementasi dan Konfigurasi Node Kontroller Pencahayaan Ruangan.....	36
3.3.4	Implementasi dan Konfigurasi Node Kontroller Pintu dan Sensor <i>Visitor Counter</i> .....	38
3.3.5	Implementasi dan Konfigurasi Node Gateway.....	39
3.3.6	Implementasi dan Konfigurasi Aplikasi Android.....	41
3.4	Uji Coba Sistem.....	43
3.4.1	Uji Coba Node Kontroller Temperatur.....	44
3.4.2	Uji Coba Node Kontroller Pencahayaan Ruangan .....	44
3.4.3	Uji Coba Node Kontroller Pintu dan Sensor <i>Visitor Counter</i> .....	45
3.4.4	Uji Coba Aplikasi Android.....	45
3.5	Skenario Pengukuran .....	46
3.5.1	Pengukuran Sensor Temperatur Ruangan .....	46
3.5.2	Pengukuran Sensor Intensitas Cahaya.....	47
3.5.3	Pengukuran Sensor <i>Visitor Counter</i> .....	47
3.5.4	Pengukuran Keandalan Kontroller Temperatur.....	48
3.5.5	Pengukuran Keandalan Kontroller Pintu.....	49
3.5.6	Pengukuran Keandalan Kontroller Lampu .....	49
3.5.7	Pengukuran Delay Pengiriman Data Antar Node.....	49
3.5.8	Pengukuran Delay Pengiriman Data ke Server IoT.....	50

3.5.9	Pengukuran Delay Pengiriman Perintah Dari Android ke Server IoT.....	50
3.5.10	Pengukuran Delay Pengambilan Data Server IoT dari Android ..	51
3.5.11	Pengukuran Delay Keseluruhan Sistem.....	51
<b>BAB 4 ANALISIS KINERJA SISTEM.....</b>		<b>53</b>
4.1	Analisis Pengukuran Sensor Temperatur Ruangan .....	53
4.2	Analisis Pengukuran Sensor Intensitas Cahaya.....	54
4.3	Analisis Pengukuran Sensor <i>Visitor Counter</i> .....	56
4.4	Analisis Keandalan Kontroller Temperatur .....	58
4.5	Analisis Keandalan Kontroller Pintu .....	61
4.6	Analisis Keandalan Kontroller Lampu.....	61
4.7	Analisis Pengukuran Delay Pengiriman Data Antar Node.....	64
4.8	Analisis Pengukuran Delay Pengiriman Data ke Server IoT ..	65
4.9	Analisis Pengukuran Delay Pengiriman Perintah Dari Android ke Server IoT .....	67
4.10	Analisis Pengukuran Delay Pengambilan Data Server IoT dari Android.....	68
4.11	Analisis Pengukuran Delay Keseluruhan Sistem .....	70
<b>BAB 5 KESIMPULAN DAN SARAN .....</b>		<b>73</b>
5.1	Kesimpulan .....	73
5.2	Saran .....	74
<b>DAFTAR PUSTAKA .....</b>		<b>77</b>
<b>LAMPIRAN.....</b>		<b>79</b>
A.	Lembar Pengesahan Proposal .....	79
B.	Foto Node .....	80
C.	Tampilan Keseluruhan Aplikasi Android .....	80
D.	Instalasi Node.....	84
E.	Tampilan Grafik ThingSpeak.....	86
F.	Source Code Program .....	89
<b>RIWAYAT HIDUP .....</b>		<b>91</b>



## DAFTAR GAMBAR

<b>Gambar 2.1</b> Cakupan Internet of Things .....	6
<b>Gambar 2.2</b> Lapisan OSI .....	9
<b>Gambar 2.3</b> Frame data UDP .....	11
<b>Gambar 2.4</b> Blok Diagram ATMEL AVR Mikrokontroler .....	13
<b>Gambar 2.5</b> Arduino Nano R3.....	14
<b>Gambar 2.6</b> Intel Galileo GEN 2 .....	15
<b>Gambar 2.7</b> ESP8266 Wifi Module.....	16
<b>Gambar 2.8</b> Sensor DHT22 .....	18
<b>Gambar 2.9</b> Sensor LDR .....	19
<b>Gambar 2.10</b> Sensor LDR dengan resistor pembagi tegangan .....	19
<b>Gambar 2.11</b> Sensor Infrared Proximity E18-D80NK .....	20
<b>Gambar 2.12</b> Kunci Pintu Elektrik LY-03.....	21
<b>Gambar 2.13</b> Rangkaian pengendali lampu.....	22
<b>Gambar 3.1</b> Diagram Alir Perancangan sistem .....	23
<b>Gambar 3.2</b> Blok diagram sistem .....	24
<b>Gambar 3.3</b> Diagram Alir Node .....	26
<b>Gambar 3.4</b> Diagram Alir Aplikasi Telepon Pintar .....	27
<b>Gambar 3.5</b> Rangkaian Kontroller Temperatur .....	28
<b>Gambar 3.6</b> Blok Kontroller Pintu dan Sensor Visitor Counter .....	29
<b>Gambar 3.7</b> Blok Node Kontroller Pencahayaan Ruangan .....	29
<b>Gambar 3.8</b> Blok Node Gateway.....	30
<b>Gambar 3.9</b> Rancangan Blok Tampilan Android .....	31
<b>Gambar 3.10</b> Membuat Channel.....	32
<b>Gambar 3.11</b> Konfigurasi Channel .....	32
<b>Gambar 3.12</b> <i>API Key</i> .....	33
<b>Gambar 3.13</b> Membuat <i>TalkBack</i> .....	33
<b>Gambar 3.14</b> Konfigurasi <i>TalkBack</i> .....	34
<b>Gambar 3.15</b> <i>TalkBack ID</i> dan <i>API Key</i> .....	34
<b>Gambar 3.16</b> Wujud Node Kontroller Temperature .....	35
<b>Gambar 3.17</b> Potongan Program Node Kontroller Temperature .....	36
<b>Gambar 3.18</b> Node Kontroller Pencahayaan Ruangan .....	37
<b>Gambar 3.19</b> Potongan program Node Kontroller Pencahayaan Ruangan .....	37
<b>Gambar 3.20</b> Node Kontroller Pintu dan <i>Visitor Counter</i> .....	38

<b>Gambar 3.21</b>	Potongan Program Untuk Menghitung Jumlah Orang ...	39
<b>Gambar 3.22</b>	Potongan Program Untuk Mengunggah Data ke Server .....	40
<b>Gambar 3.23</b>	Node Gateway .....	41
<b>Gambar 3.24</b>	(a). Tampilan Utama Aplikasi (b). Tampilan Saat Menu di Buka.....	42
<b>Gambar 3.25</b>	Tampilan (a). Temperatur Ruangan (b). Pencahayaan ruangan Manual (c). Pencahayaan Ruangan Otomatis .....	42
<b>Gambar 3.26</b>	Potongan Program Pengambilan Data dari Server .....	43
<b>Gambar 3.27</b>	Uji Coba Node Kontroller Temperatur.....	44
<b>Gambar 3.28</b>	Uji Coba Node Kontroller Pencahayaan Ruangan .....	44
<b>Gambar 3.29</b>	Uji Coba Node Kontroler Pintu dan Sensor <i>Visitor Counter</i> .....	45
<b>Gambar 3.30</b>	Uji Coba (a). Ambil Data dari Server (b). Membuat Grafik Data (c). Kontrol Node .....	46
<b>Gambar 3.31</b>	Skenario Pengukuran Sensor Temperatur Ruangan .....	47
<b>Gambar 3.32</b>	Skenario Pengukuran Sensor Intensitas Cahaya.....	47
<b>Gambar 3.33</b>	Skenario Pengukuran Sensor <i>Visitor Counter</i> .....	48
<b>Gambar 3.34</b>	Skenario Pengukuran Delay Pengiriman Data antar Node .....	50
<b>Gambar 3.35</b>	Skenario Pengukuran Delay Pengiriman Data ke Server.....	50
<b>Gambar 3.36</b>	Skenario Pengukuran Delay Pengiriman Data ke Server dari Aplikasi Android .....	51
<b>Gambar 3.37</b>	Skenario Pengukuran Delay Pengiriman Data ke Server.....	51
<b>Gambar 3.38</b>	Skenario Pengukuran Delay Keseluruhan Sistem .....	52
<b>Gambar 4.1</b>	Grafik Pengukuran Sensor Suhu dengan Termometer .....	54
<b>Gambar 4.2</b>	Grafik Pengukuran Intensitas Cahaya Vs Jarak.....	56
<b>Gambar 4.3</b>	Grafik Pengukuran <i>Visitor Counter</i> Arah Masuk Ruangan .....	57
<b>Gambar 4.4</b>	Grafik Pengukuran <i>Visitor Counter</i> Arah Keluar Ruangan .....	58
<b>Gambar 4.5</b>	Grafik Pengukuran Delay Pengiriman Data Antar Node Sensor Dengan Node Gateway .....	65
<b>Gambar 4.6</b>	Grafik Pengukuran Delay Pengiriman Data dari Node Ke Server IoT .....	66
<b>Gambar 4.7</b>	Grafik Pengukuran Delay Pengiriman Data dari Android ke Server IoT .....	68

<b>Gambar 4.8</b> Grafik Pengukuran Delay Pengambilan Data Server IoT dari Android .....	69
<b>Gambar 4.9</b> Grafik Pengukuran Delay Keseluruhan Sistem .....	71

## DAFTAR TABEL

<b>Tabel 2.1</b> Spesifikasi Arduino Nano R3 .....	14
<b>Tabel 2.2</b> Spesifikasi Intel Galileo .....	15
<b>Tabel 2.3</b> Spesifikasi Modul Wifi ESP8266.....	17
<b>Tabel 2.4</b> Spesifikasi Sensor DHT22 .....	18
<b>Tabel 2.5</b> Spesifikasi Sensor Infrared Proximity E18-D80NK.....	21
<b>Tabel 2.6</b> Spesifikasi Kunci Pintu Elektrik .....	22
<b>Tabel 4.1</b> Hasil Pengukuran Sensor Temperatur dan Termometer.....	53
<b>Tabel 4.2</b> Hasil Pengukur Sensor Intensitas Cahaya (LDR).....	55
<b>Tabel 4.3</b> Hasil Pengukuran Sensor <i>Visitor Counter</i> .....	57
<b>Tabel 4.4</b> Hasil Pengukuran Keandalan Kontroller Tempertur Mode Otomatis.....	59
<b>Tabel 4.5</b> Hasil Pengukuran Keandalan Kontroller Tempertur Mode Manual .....	60
<b>Tabel 4.6</b> Hasil Pengukuran Keandalan Kontroller Pintu.....	61
<b>Tabel 4.7</b> Hasil Pengukuran Keandalan Kontroller Lampu mode otomatis .....	62
<b>Tabel 4.8</b> Hasil Pengukuran Keandalan Kontroller Lampu Mode Manual .....	63
<b>Tabel 4.9</b> Hasil Pengukuran Delay Pengiriman Data Antar Node Sensor Dengan Node Gateway .....	64
<b>Tabel 4.10</b> Hasil Pengukuran Delay Pengiriman Data dari Node Ke Server <i>IoT</i> .....	66
<b>Tabel 4.11</b> Hasil Pengukuran Delay Pengiriman Data dari Android Ke Server <i>IoT</i> .....	67
<b>Tabel 4.12</b> Hasil Pengukuran Delay Pengambilan Data Server <i>IoT</i> dari Android .....	69
<b>Tabel 4.13</b> Hasil Pengukuran Delay Keseluruhan Sistem .....	70

# **BAB 1**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Saat ini kebutuhan Manusia akan energi listrik dan mobilitas sangat tinggi. Boleh dikatakan tanpa listrik kita tidak bisa hidup dengan nyaman. Hampir seluruh peralatan rumah tangga memerlukan listrik. Mulai dari TV, komputer, pompa air, rice cooker, AC dan alat rumah tangga lainnya. saat ini energi listrik terbilang mahal terlebih dengan adanya global warming yang mengancam kehidupan manusia. Untuk menghemat pemakaian energi listrik saat ini banyak pengembang perumahan menerapkan teknologi Smart home atau Smart Office yang bertujuan untuk lebih mengefisiensikan pemakaian listrik di rumah. Pada sistem Smart home atau Smart Office, penggunaan listrik akan diatur sedemikian rupa sehingga meningkatkan efisiensi penggunaan listrik.

Selain untuk mengefisienkan penggunaan energy, Smart home atau Smart Office juga menjadi solusi kebutuhan akan kemudahan mengatur dan memonitoring perangkat listrik ataupun perangkat lain di rumah ataupun kantor dengan cara mengintegrasikan perangkat Smart home atau Smart Office dengan perangkat gadget atau smartphone yang notabene hampir semua orang saat ini memilikinya dan selalu dibawa kemanapun.

Konsep Smart home atau Smart Office pada umumnya membatasi pengguna harus berada pada satu jaringan local yang sama dalam melakukan monitoring dan pengaturan perangkat. Namun dengan konsep Internet of Things keterbatasan tersebut bias di hilangkan.

Internet of Things, atau dikenal juga dengan singkatan IoT, merupakan sebuah konsep yang bertujuan untuk memperluas manfaat dari konektivitas internet yang tersambung secara terus-menerus, seperti berbagi data, remote control, monitoring sistem, dan sebagainya.

### **1.2 Rumusan Masalah**

Permasalahan yang dibahas dalam tugas akhir ini adalah:

1. Bagaimana kinerja sistem kantor pintar yang telah dibuat?.
2. Bagaimana sensitifitas respon sensor yang digunakan dalam sistem kantor pintar?.
3. Bagaimana delay pengiriman mempengaruhi kinerja dari sistem kantor pintar?.

### 1.3 Batasan Masalah

Batasan masalah dari Tugas Akhir ini adalah:

1. Pengujian sistem akan dilakukan di LAB B301 T.Elektro ITS.
2. Komunikasi antara perangkat sensor ke gateway menggunakan wifi 802.11x.
3. Protocol komunikasi yang digunakan untuk mengakses data server adalah http request method, sedangkan untuk komunikasi antar node adalah protokol UDP.

### 1.4 Tujuan

Tujuan dan manfaat dari Tugas Akhir ini adalah untuk memenuhi kebutuhan akan kemudahan dan kenyamanan mengontrol dan memonitoring peralatan dan perangkat di suatu kantor pintar ataupun rumah pintar melalui teknologi internet.

### 1.5 Metodologi

Metodologi yang digunakan dalam tugas akhir ini terdiri dari beberapa tahap sebagai berikut:

1. Studi literatur  
Mencari dan mempelajari beberapa buku, paper, jurnal maupun literatur ilmiah lainnya mengenai teori maupun mengenai hal-hal teknis lain yang dibutuhkan untuk menyelesaikan tugas akhir. Selain itu juga melakukan *internet* research. Pada tahap ini akan dipelajari cara kerja serta konfigurasi dari *Internet of Things* serta hal-hal yang terkait dengan perancangan *Internet of Things Smart Office*.
2. Mengumpulkan Data  
Pada tahapan ini pengumpulan data yaitu pengambilan sampel data dimulai dari data temperatur ruangan ideal, intensitas cahaya ruangan yang ideal dan data-data lainnya.
3. Perencanaan *Hardware Smart Office*  
Pada tahap ini dilakukan perancangan dan pembuatan hardware, dimulai dengan mendesain rangkaiannya kemudian survei komponen di pasaran baru kemudian pembuatan hardware dapat dilakukan.
4. Perancangan *Software Smart Office*  
Tahap ini baru bisa dilakukan ketika perancangan dan pembuatan hardware telah selesai, dimana pada tahapan ini hardware yang telah dibuat diprogram sesuai dengan fungsinya masing-masing.

5. Analisis Data  
Pada tahap ini dilakukan pengamatan terhadap hasil data yang dihasilkan pada tahap perancangan untuk dianalisis lebih lanjut lagi.
6. Penarikan Kesimpulan  
Analisis data yang telah didapatkan akan digunakan untuk penarikan kesimpulan.

## 1.6 Sistematika Pembahasan

Pembahasan dalam tugas akhir ini akan dibagi dalam lima bab dengan sistematika sebagai berikut:

### **Bab I Pendahuluan**

Pada bab ini dijelaskan mengenai latar belakang, rumusan masalah, batasan masalah, tujuan, metodologi, sistematika pembahasan dan relevansi.

### **Bab II *IoT Smart Office* dan Teori Penunjang Perancangannya**

Pada bab ini dijelaskan mengenai tinjauan pustaka mengenai *IoT Smart Office* baik dari sisi konsep secara umum, komponennya maupun tahapan untuk pembangunan *IoT Smart Office*. Selain itu juga dijelaskan tinjauan pustaka terkait teknologi kantor pintar dengan memanfaatkan *Internet of Things*.

### **Bab III Perancangan Sistem Kantor Pintar**

Pada bab ini dijelaskan hal-hal dan data-data yang berkaitan secara langsung dengan perancangan sistem kantor pintar yang akan dibuat.

### **Bab IV Analisis Kinerja Sistem**

Bab ini berisi pengolahan data perancangan dari *IoT Smart Office* yang dibangun dimana data yang didapat dianalisis untuk memudahkan menarik kesimpulan.

### **Bab V Kesimpulan dan Saran**

Pada bab ini berisi kesimpulan dan saran.

## **1.7 Relevansi**

Hasil dari yang didapat pada tugas akhir ini diharapkan dapat memberi manfaat dan solusi sekaligus sebagai referensi untuk implementasi teknologi *Smart Office* berbasis *Internet of Things*. Hal ini tentu erat kaitannya dengan kebutuhan akan kemudahan dan kenyamanan akses perangkat elektronik pada Kantor guna menunjang akses informasi dan wacana intelektual dalam dunia modern yang perkembangannya semakin cepat dan luas.



## **BAB 2**

### ***IOT SMART OFFICE* DAN TEORI PENUNJANG PERANCANGANNYA**

#### **2.1 Konsep Smart Office**

Konsep kantor pintar (*smart office*) bukan lagi impian. Kantor berisi perangkat multifungsi dan terkoneksi dengan pemilik kantor ataupun karyawan, kini sudah mulai diterapkan secara global. *Smart office* pada dasarnya adalah kantor biasa. Namun, dikatakan pintar karena ada perangkat-perangkat cerdas di dalam kantor tersebut. Ketika semua perangkat terhubung menjadi satu, itulah *smart office*.

Dengan konsep *smart office* / *smart home* memungkinkan pemilik rumah / karyawan untuk melakukan atau mengontrol berbagai aktivitas dengan menggunakan ponsel pintar, menjadi salah satu hal yang dikedepankan dalam konsep *smart office*/ *smart home*.

*Smart office* / *smart home* adalah konsep yang sedang ramai diperbincangkan baik di dalam negeri maupun di luar negeri. Konsep tempat seperti ini mulai dikenal banyak orang sejak pertama kali dicetuskan oleh sebuah perusahaan milik Joe Dada dan David Tan. Perusahaan ini berdiri pada tahun 1992 dengan menyediakan sistem elektronik home automation dan menjual segala perangkat pendukungnya melalui [www.smarthome.com](http://www.smarthome.com). Dengan perangkat canggih yang saling terkoneksi, pemilik rumah dapat mengatur mulai dari suhu ruangan, home theater, hingga kamera keamanan di manapun mereka berada. Sebenarnya, teknologi-teknologi tersebut telah banyak diterapkan di industri, perkantoran, pusat bisnis, namun belum banyak diterapkan pada rumah pribadi. Teknologi yang paling umum diterapkan di sebuah rumah adalah teknologi pengamanan dalam bentuk kamera CCTV (Closed-circuit television), yang bisa meminimalisasi ancaman pencurian.

Kini teknologi yang diterapkan pada tempat tinggal dan atau kantor mulai semakin beragam. Segalanya pun menjadi serba praktis dan mudah. Ingin menghidupkan televisi, atau mematikan lampu tanpa harus bangun dari tempat duduk, Anda bisa melakukannya dengan hanya menekan tombol pada ponsel Anda.

Sejumlah perusahaan teknologi seperti Google juga mulai mengembangkan produk mereka untuk memenuhi permintaan pasar akan rumah pintar. Contohnya, seperti yang diwartakan The Wall Street pada

bulan Januari 2014, Google merogoh kocek yang sangat besar jumlahnya yaitu USD 3,2 miliar atau sekitar Rp43,3 triliun untuk membeli sebuah perusahaan pembuat rumah tangga pintar yaitu Nest Labs. Nest Labs merupakan perusahaan pembuat alarm asap dan pengatur suhu.

## 2.2 Internet of Things

Internet of things (IoT) adalah istilah yang menggambarkan bagaimana berbagai perangkat di sekitar kita selain bisa terhubung ke internet, juga dapat berkomunikasi dengan tablet, PC, dan smartphone. Adalah Kevin Ashton yang pertama mencetus istilah IoT pada 1999. Ashton, si pencetus standar global untuk RFID itu sudah lama membayangkan bahwa internet dapat terhubung ke dunia fisik melalui berbagai sensor yang ditanamkan di perangkat tertentu, mengumpulkan data, untuk lantas mengirimkannya ke database atau log server.

Gambar 2.1 dibawah ini menggambarkan cakupan dari konsep Internet of Things dimana konsep ini menawarkan kemudahan dan kenyamanan dalam mengakses perngkat kita dari mana saja dan kapan saja.



**Gambar 2.1** Cakupan Internet of Things

Penerapan IoT tiga tahun kedepan akan lebih terlihat kepada aspek seperti smart home, atau alat pemantau tubuh manusia. Berbagai perangkat yang terkoneksi tersebut tidak butuh layar dan kibor untuk beroperasi. Tidak perlu pula dicolok ke komputer desktop untuk mengunduh datanya. Sebab, smartphone dan tablet akan menjadi hub atau pusat untuk mengontrol berbagai perangkat tersebut.

Contoh sehari-hari yang dapat kita lihat seperti gelang Fitbit, yang mencatat aktivitas fisik penggunanya. Semua informasi yang telah direkam lantas dikirim ke aplikasi smartphone untuk disimpan dan diproses.

### **2.3 Aplikasi Server Internet of Things**

*Server* merupakan sebuah sistem komputer yang menyediakan jenis layanan tertentu dalam sebuah jaringan komputer. *Server* didukung dengan prosesor yang bersifat scalable dan *RAM* yang besar, dan juga dilengkapi dengan sistem operasi khusus, yang disebut sebagai sistem operasi jaringan. *Server* juga menjalankan perangkat lunak administratif yang mengontrol akses terhadap jaringan dan sumber daya yang terdapat di dalamnya contoh seperti halnya berkas atau data lainnya, dan memberikan akses kepada pengguna. *Server Internet of Things* dalam tugas akhir ini menggunakan aplikasi server open source yang dikembangkan oleh Mathworks yaitu ThingSpeak Server. Dengan ThingSpeak kita dapat melakukan 3 Hal yang sangat berguna untuk menampung dan menganalisis data, 3 hal tersebut yaitu:

#### *a. Collect*

Mengirimkan / mengonlinekan data-data sensor ke Internet (Thingspeak Server). Thingspeak menyediakan API untuk mengakses semua fitur yang disediakan. Sebelum dapat melakukan pengumpulan data, terlebih dahulu kita harus punya akun dan channel di thingspeak, Akun digunakan untuk Login ke Thingspeak, Channel fungsinya sama halnya Tabel pada MySQL. Setelah membuat akun dan channel kita akan diberikan Write API KEY dan Read API KEY yang berguna sebagai identitas saat mengirimkan data ke ThingSpeak atau membaca data dari ThingSpeak. Semua komunikasi dengan ThingSpeak dilakukan dengan HTTP Method.

#### *b. Analyze*

Menganalisa dan memvisualisasikan data. Dengan menyimpan data di Online Storage akan memudahkan dalam mengakses data, ThingSpeak menyediakan Tools untuk melakukan analisis dan visualisasi data secara Online. ThingSpeak juga support dengan program MATLAB. (program Matlab dapat di Jalankan di Server ThingSpeak dan juga bisa dijalankan di Desktop PC kita). Data hasil analisa dapat disajikan dalam bentuk plot, grafik atau Gauge.

### c. *ACT*

Menttrigger suatu perintah / kegiatan berdasarkan data hasil analisa. Dengan fitur ACT kita dapat menttrigger suatu aksi apabila suatu kondisi hasil analisa terpenuhi

Misal: saat temperature ruangan mencapai  $> 30$  derajat maka akan menttrigger agar kipas menyala. Untuk bisa melakukannya terlebih dahulu harus mengkonfigurasi TalkBackApp di Thingspeak.

## 2.4 **Aplikasi Android**

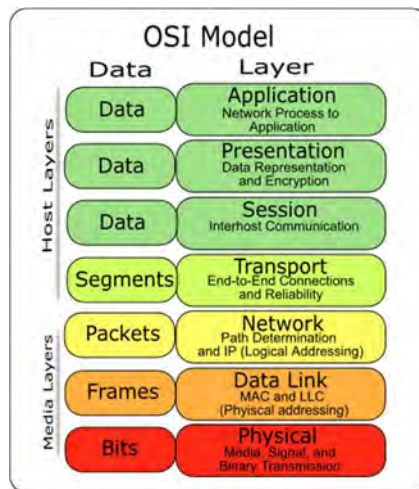
Aplikaasi Android adalah sebuah kumpulan perangkat lunak untuk perangkat mobile yang mencakup sistem operasi, middleware dan aplikasi utama mobile. Android sendiri adalah sebuah sistem operasi untuk perangkat mobile berbasis linux yang mencakup sistem operasi, middlewere dan aplikasi. Android menyediakan platform terbuka bagi para pengembang untuk membuat aplikasi mereka. Awalnya, Google Inc. membeli Android Inc. yang merupakan pendatang baru yang membuat software untuk ponsel/smartphone. Kemudian untuk mengembangkan Android, dibentuklah *Open Handset Alliance*, konsorsium dari 34 perusahaan hardware, software, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nidia. Pada saat perilis perdana Android, 5 November 2007, Android bersama *Open Handset Alliance* menyatakan mendukung pengembangan open source pada perangkat mobile. Di pihak lain, Google merilis kode-kode Android di bawah lisensi Apache, sebuah lisensi software dan open *paltform* perangkat seluler. Pada masa saat ini sebagian besar vendor-vendor smartphone sudah memproduksi smartphone berbasis Android, seperti HTC, Motorola, Samsung, LG dan masih banyak vendor lainnya. Hal ini disebabkan karena Android adalah sistem operasi yang *open source* sehingga bebas didistribusikan dan dipakai oleh vendor manapun.

## 2.5 **Protokol Komunikasi**

Untuk dapat berkomunikasi antara perangkat maka dibutuhkan suatu protokol komunikasi. Protokol komunikasi adalah suatu tatacara yang digunakan untuk melaksanakan pertukaran data (pesan) antara duabuah system dalam jaringan. Kedua system bias saja berbeda sama sekali, dan protocol ini mengurus perbedaan format pada keduanya.

### 2.5.1 Lapisan OSI

Standar protocol yang terkenal yaitu OSI (*Open System Interconnection*) yang dibuat oleh ISO (International for Standarization Organization). OSI mengidentifikasi semua proses yang dibutuhkan untuk melakukan komunikasi dan membaginya ke dalam kelompok secara logika yang disebut layer. Gambar 2.2 dibawah ini merupakan gambaran jelas dari susunan layer OSI.



**Gambar 2.2** Lapisan OSI

OSI terdiri dari tujuh layer, yang secara umum terbagi dalam dua kelompok, yakni *Upper layer* (*Application Layer*) dan *Lower layer* (*Data Transport Layer*). Layer yang tergolong dalam *upper layer* mendefinisikan bagaimana aplikasi pada sebuah host akan berkomunikasi dengan user dan host lainnya. Sedangkan *lower layer* mendefinisikan bagaimana data terkirim dari satu host ke host lainnya.

### 2.5.2 Protokol HTTP

HTTP menetapkan sembilan metode (kadang disebut "verbs") yang menunjukkan tindakan yang ingin dilakukan terhadap sumber teridentifikasi. Hal yang diwakili sumber ini, berupa data yang sudah ada atau data yang diciptakan secara dinamis, bergantung pada implementasi

server. Biasanya sumber ini berkaitan dengan berkas atau keluaran dari berkas pelaksana yang menetap di server.

a. *HEAD*

Meminta tanggapan yang identik dengan tanggapan yang sesuai dengan permintaan *GET*, namun tanpa badan tanggapan. Ini berguna untuk mengakses informasi meta yang tertulis dalam kepala tanggapan tanpa perlu mengangkut seluruh konten.

b. *GET*

Meminta representasi sumber tertentu. Permintaan menggunakan *GET* (dan beberapa metode *HTTP* lain) "tidak boleh memiliki kepentingan melakukan tindakan selain pengaksesan". W3C telah menerbitkan prinsip panduan mengenai perbedaan ini dengan menyatakan, "desain aplikasi web harus mematuhi prinsip di atas, serta batasan sejenis."

c. *POST*

Mengirimkan data untuk diproses (misalnya dari bentuk HTML) ke sumber teridentifikasi. Data dimasukkan dalam badan permintaan. Ini dapat menghasilkan pembentukan sumber baru atau pemutakhiran sumber yang sudah ada atau keduanya.

d. *PUT*

Mengunggah representasi sumber tertentu.

e. *DELETE*

Menghapus sumber tertentu.

f. *TRACE*

Mengaungkan kembali permintaan yang diterima, sehingga klien dapat melihat perubahan atau tambahan yang dilakukan oleh peladen perantara.

g. *OPTIONS*

Mengembalikan metode *HTTP* yang didukung server untuk URL tertentu. Ini dapat digunakan untuk memeriksa fungsionalitas peladen web dengan meminta '\*' daripada fungsionalitas sumber tertentu.

h. *CONNECT*

Menukarkan koneksi permintaan dengan terowongan TCP/IP transparan, biasanya untuk memfasilitasi komunikasi terenkripsi *SSL (HTTPS)* melalui proksi *HTTP* tak terenkripsi

i. *PATCH*

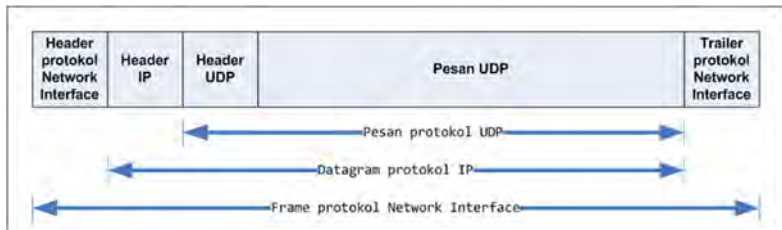
Menerapkan modifikasi parsial terhadap sumber.

### 2.5.3 Protokol UDP

UDP, singkatan dari User Datagram Protocol, adalah salah satu protokol lapisan transpor TCP/IP yang mendukung komunikasi yang tidak andal (unreliable), tanpa koneksi (connectionless) antara host-host dalam jaringan yang menggunakan TCP/IP.

#### 2.5.3.1 Frame Data UDP

UDP, berbeda dengan TCP yang memiliki satuan paket data yang disebut dengan segmen, melakukan pengepakan terhadap data ke dalam pesan-pesan UDP (UDP Messages). Gambar 2.3 berikut ini menggambarkan pengepakan data dalam frame UDP.



**Gambar 2.3** *Frame data UDP*

Dalam header IP dari sebuah pesan UDP, field Source IP Address akan diset ke antarmuka host yang mengirimkan pesan UDP yang bersangkutan; sementara field Destination IP Address akan diset ke alamat IP unicast dari sebuah host tertentu, alamat IP broadcast, atau alamat IP multicast.

#### 2.5.3.2 Karakteristik UDP

- Connectionless (tanpa koneksi): Pesan-pesan UDP akan dikirimkan tanpa harus dilakukan proses negosiasi koneksi antara dua host yang hendak bertukar informasi.
- Unreliable (tidak andal): Pesan-pesan UDP akan dikirimkan sebagai datagram tanpa adanya nomor urut atau pesan acknowledgment. Protokol lapisan aplikasi yang berjalan di atas UDP harus melakukan pemulihan terhadap pesan-pesan yang hilang selama transmisi. Umumnya, protokol lapisan aplikasi yang berjalan di atas UDP mengimplementasikan layanan keandalan mereka masing-

masing, atau mengirim pesan secara periodik atau dengan menggunakan waktu yang telah didefinisikan.

- c. UDP menyediakan mekanisme untuk mengirim pesan-pesan ke sebuah protokol lapisan aplikasi atau proses tertentu di dalam sebuah host dalam jaringan yang menggunakan TCP/IP. Header UDP berisi field Source Process Identification dan Destination Process Identification.
- d. UDP menyediakan penghitungan checksum berukuran 16-bit terhadap keseluruhan pesan UDP.

#### **2.5.3.3 Layanan antar host yang tidak disediakan UDP**

- a. UDP tidak menyediakan mekanisme penyanggaan (buffering) dari data yang masuk ataupun data yang keluar. Tugas buffering merupakan tugas yang harus diimplementasikan oleh protokol lapisan aplikasi yang berjalan di atas UDP.
- b. UDP tidak menyediakan mekanisme segmentasi data yang besar ke dalam segmen-segmen data, seperti yang terjadi dalam protokol TCP. Karena itulah, protokol lapisan aplikasi yang berjalan di atas UDP harus mengirimkan data yang berukuran kecil (tidak lebih besar dari nilai Maximum Transfer Unit/MTU) yang dimiliki oleh sebuah antarmuka di mana data tersebut dikirim. Karena, jika ukuran paket data yang dikirim lebih besar dibandingkan nilai MTU, paket data yang dikirimkan bisa saja terpecah menjadi beberapa fragmen yang akhirnya tidak jadi terkirim dengan benar.
- c. UDP tidak menyediakan mekanisme flow-control, seperti yang dimiliki oleh TCP.

### **2.6 Sistem Tertanam (*Embedded System*)**

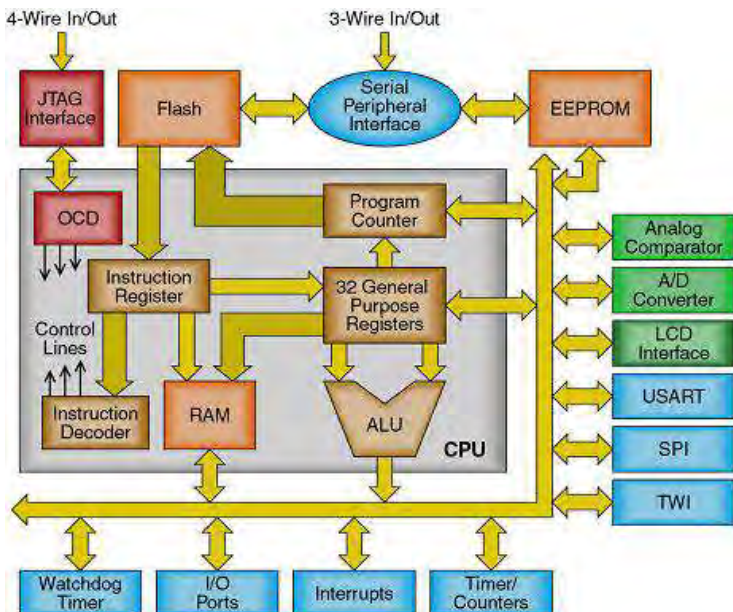
Sistem embedded biasanya mengandung syarat minimal sebuah sistem mikroprosesor yaitu memori untuk data dan program, serta sistem antarmuka input/output yang sederhana. Antarmuka semacam keyboard, tampilan, disket, atau printer yang umumnya ada pada sebuah komputer pribadi justru tidak ada pada sistem mikrokontroler. Sistem mikrokontroler lebih banyak melakukan pekerjaan-pekerjaan sederhana yang penting seperti mengendalikan motor, saklar, resistor variabel, atau perangkat elektronis lain. Seringkali satu-satunya bentuk antarmuka yang ada pada sebuah sistem mikrokontroler hanyalah sebuah LED, bahkan ini pun bisa dihilangkan jika tuntutan konsumsi daya listrik mengharuskan



demikian. Saat ini banyak sekali jenis mikrokontroller ataupun mikroprosesor embedded yang beredar di dunia

### 2.6.1 AVR Mikrokontroller

AVR adalah perangkat yang di desain untuk berjalan dengan cepat, dengan menggunakan instruksi mesin yang disederhanakan sehingga dapat meningkatkan kinerja dari mikrokontroller. (sebelum ada RISC, namanya *CISC(Complex Instruction Set Computers)*). dengan penggunaan instruksi yang lebih sederhana memberikan kontribusi pada kecepatan dengan instruksi mesin yang terbatas. Mikrokontroller AVR RISC dapat berjalan pada single cycle dari prosesor clock, yang berarti Mikrokontroller AVR dengan clock 8 MHz, dapat mengeksekusi sekitar 8 juta instruksi perdetiknya atau 8 MIPS(million instruction per second). Gambar 2.4 berikut ini menjelaskan arsitektur dalam mikrokontroller AVR.



**Gambar 2.4** Blok Diagram ATMEGA AVR Mikrokontroller

AVR juga dikembangkan oleh Arduino.cc sehingga menjadi platform baru berlabel Arduino, pada dasarnya AVR dan Arduino adalah sama, hanya saja yang membedakan adalah pada Arduino telah ditanamkan bootloader yang memungkinkan Arduino diidi program melalui interface Serial, dan Arduino menyediakan berbagai macam library untuk mengakses berbagaimacam sensor dan aktuator secara gratis. Hal inilah yang membuat arduino lebih dikenal dari pada AVR. Gambar 2.5 merupakan gambar bentuk fisik dari arduino nano R3 yang memiliki spesifikasi lengkap seperti ditunjukkan pada Tabel 2.1 dibawah ini.



**Gambar 2.5** Arduino Nano R3

**Tabel 2.1** Spesifikasi Arduino Nano R3

Microcontroller	ATmega328
Voltage	5 V
Input Voltage	7-12 V
Input Voltage (limits)	6-20 V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input	8
Current per I/O	40 mA
Flash Memory	16 KB (ATmega168) or 32 KB (ATmega328) of which 2 KB used by bootloader
SRAM	1 KB (ATmega168) or 2 KB (ATmega328)
EEPROM	512 bytes (ATmega168) or 1 KB (ATmega328)
Clock Speed	16 MHz

### 2.6.2 Single Board Computer (SBC)

Komputer papan tunggal (*Single Board Computer / SBC*) yang memiliki ukuran kecil. *Single Board Computer* bisa digunakan untuk berbagai keperluan, seperti spreadsheet, game, bahkan bisa digunakan sebagai media player karena kemampuannya dalam memutar video high definition. Khusus untuk *Single Board* yang digunakan untuk aplikasi *embedded system* biasanya dilengkapi dengan beberapa I/O (*Input Output*) *peripheral* dengan tujuan untuk mempermudah melakukan *interfacing* dengan perangkat lain seperti sensor ataupun aktuator. Beberapa *Single Board Computer* yang beredar dan dipakai saat ini adalah Intel Galileo Gen2. Gambar 2.6 merupakan gambar bentuk fisik dari Intel Galileo Gen2 yang memiliki spesifikasi lengkap seperti ditunjukkan pada Tabel 2.2.



**Gambar 2.6** Intel Galileo GEN 2

**Tabel 2.2** Spesifikasi Intel Galileo

Microcontroller	SoC Quark X1000
Operating Voltage	3.3V / 5V
Flash Memory	512 kB
RAM	256 MB DDR3
SRAM	512 kB
Flash Storage	8MB
EEPROM	8kB
Clock Speed	400 MHz
PoE compatible	Yes

Secara umum, Intel Galileo adalah Embedded board yang merupakan gabungan dari Arduino dengan processor Intel Quark SoC X1000. Intel Galileo berbasis prosessor X86 Intel Quark SoC X1000 yang memiliki performa tinggi dibandingkan arduino tipe lainnya. Board ini menjalankan sistem operasi linux distro Yocto Clanton. Board ini memiliki interface Ethernet dan 2 buah port USB. Dan yang menjadi unggulan dari board ini adalah tersedianya konektor yang sama persis pinoutnya dengan arduino uno sehingga membuat board ini kompatibel dengan berbagai macam shield yang dapat digunakan di arduino uno.

### 2.6.3 Modul Wifi

Wifi modul adalah perangkat radio wifi yang dapat digunakan untuk memberikan akses koneksi wireless kepada mikrokontroler supaya bisa terkoneksi ke jaringan. ESP8266 WiFi Module adalah wifi modul SOC yang terintegrasi dengan TCP / IP stack protokol yang dapat memberikan akses ke mikrokontroler jaringan WiFi. Setiap modul ESP8266 kondisi default pra-diprogram dengan perintah AT set firmware, yang berarti kita hanya dapat menghubungkan modul ini ke perangkat Arduino kita dan mendapatkan fungsi dari wifi shield namun dengan biaya rendah. Gambar 2.6 merupakan gambar bentuk fisik dari modul wifi ESP8266 yang dipakai dalam tugas akhir ini.



**Gambar 2.7** ESP8266 Wifi Module

Modul ini memiliki *on board processing* dan penyimpanan kemampuan yang kuat yang memungkinkan untuk diintegrasikan dengan sensor dan perangkat khusus aplikasi lain melalui GPIO dengan pengembangan minimal dan minimal loading runtime. Chip terintegrasi tingkat tinggi meminimalkan penggunaan sirkuit eksternal, termasuk

modul *front-end*, dirancang untuk menempati daerah PCB minimal. ESP8266 mendukung RSDP untuk aplikasi VoIP, berisi RF yang dikalibrasi memungkinkan untuk bekerja di bawah semua kondisi operasi, dan tidak memerlukan bagian RF eksternal. modul ini adalah solusi *high performance low cost IOT (Internet of Things)*. Spesifikasi dari modul wifi ESP8266 ditunjukkan dalam Tabel 2.3 berikut.

**Tabel 2.3** Spesifikasi Modul Wifi ESP8266

Mode	802.11 b/g/n
Configuration	Wi-Fi Direct (P2P), soft-AP
TCP/IP protocol stack	Integrated TCP/IP protocol stack
Output Power	+19.5dBm in 802.11b mode
Flash Memory	1MB
Interface	SDIO 1.1 / 2.0, SPI, UART
Response Time	Wake up and transmit packets in < 2ms
Standby power consumption	< 1.0mW

#### 2.6.4 Sensor

Sensor adalah sesuatu yang digunakan untuk mendeteksi adanya perubahan lingkungan fisik atau kimia. Variabel keluaran dari sensor yang diubah menjadi besaran listrik disebut Transduser.

##### 2.6.4.1 Sensor Temperatur dan Kelembaban

Sensor temperatur dan kelembaban adalah sensor untuk mengetahui kondisi suhu dan kelembaban pada suatu area tertentu, salah satu dari sensor temperatur dan kelembaban adalah DHT22, dimana sensor digital ini dapat mengukur suhu dan kelembaban udara di sekitarnya. Sensor ini sangat mudah digunakan bersama dengan Arduino. Memiliki tingkat stabilitas yang sangat baik serta fitur kalibrasi yang sangat akurat. Koefisien kalibrasi disimpan dalam OTP program memory, sehingga ketika internal sensor mendeteksi sesuatu, maka module ini menyertakan koefisien tersebut dalam kalkulasinya. Modul sensor DHT22 ditunjukkan oleh Gambar 2.8 berikut ini.



**Gambar 2.8** Sensor DHT22

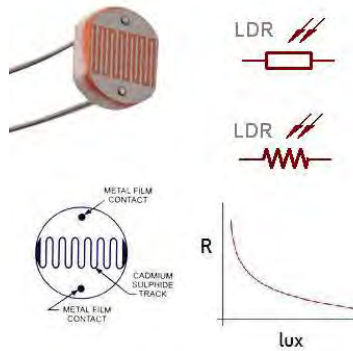
DHT22 termasuk sensor yang memiliki kualitas terbaik, dinilai dari respon, pembacaan data yang cepat, dan kemampuan anti-interferensi. Ukurannya yang kecil, dan dengan transmisi sinyal hingga 20 meter, membuat produk ini cocok digunakan untuk banyak aplikasi-aplikasi pengukuran suhu dan kelembaban, adapun spesifikasi dari sensor DHT22 ditunjukkan oleh Tabel 2.4 berikut ini:

**Tabel 2.4** Spesifikasi Sensor DHT22

Input Voltage	3.3-6V
Measuring current	1-1.5mA
Standby current	40-50 uA
Humidity	0-100% RH
Temperature range	-40 - 80 degrees C
Accuracy	+/-2% RH

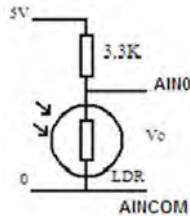
#### **2.6.4.2    *Sensor Intensitas Cahaya***

Sensor intensitas cahaya adalah sensor yang mampu mengetahui besarnya intensitas cahaya yang diterimanya. Salah satu dari jenis sensor ini adalah *Light Dependent Resistor (LDR)*. Komponen elektronik ini resistansinya akan menurun jika ada penambahan intensitas cahaya yang mengenainya. Bentuk fisik dan simbbol rangkaian dari sensor LDR ditunjukkan oleh Gambar 2.9.



**Gambar 2.9** Sensor LDR

Hasil pembacaan data sensor ini adalah berupa tegangan, untuk mengkonversi menjadi satuan Lux (lumens) dapat dilakukan dengan cara menggunakan rangkaian pada Gambar 2.10 berikut:



**Gambar 2.10** Sensor LDR dengan resistor pembagi tegangan

Resistansi dari LDR bervariasi sesuai dengan jumlah cahaya yang jatuh di atasnya. Hubungan antara resistansi  $R_L$  dan intensitas cahaya Lux untuk LDR pada umumnya adalah:

$$R_L = \frac{500}{Lux} \quad (1)$$

Jika LDR terhubung ke 5V melalui 3.3k resistor, menggunakan aturan pembagi tegangan, maka tegangan output dari LDR adalah:

$$V_o = 5 \times \frac{R_L}{R_L + 3.3} \quad (2)$$

Substitusi  $R_L$  dari persamaan 1 ke persamaan 2, kita memperoleh intensitas cahaya:

$$Lux = \frac{\frac{2500}{V_o} - 500}{3.3} \quad (3)$$

#### 2.6.4.3 *Sensor Infrared Proximity*

Proximity sensor merupakan perangkat yang mendeteksi keberadaan dan kedekatan objek baik berupa logam maupun non logam. Proximity hanya mendeteksi "keberadaan" dan tidak memberi "kuantitas" dari objek. Maksudnya, jika mendeteksi logam maka keluaran dari detektor hanya "ada" atau "tidak ada" logam. Proximity tidak memberikan informasi tentang kuantitas logam seperti jenis logam, ketebalan, jarak, suhu dll. Jadi hanya "ada atau tidak ada" logam. Juga sama untuk non logam. Proximity untuk logam biasanya dengan "inductive proximity" sedang untuk non logam dengan "capacitive proximity". Salah satu jenis sensor Infrared Proximity adalah tipe E18-D80NK seperti ditunjukkan oleh Gambar 2.11 berikut ini.



**Gambar 2.11** Sensor Infrared Proximity E18-D80NK

Sensor ini bekerja dengan cara memancarkan gelombang inframerah ke suatu objek, dan jika terdapat objek maka sensor ini akan memantulkan gelombang inframerah dan diterima lagi oleh sensor tersebut. Sensor ini dapat digunakan untuk mendeteksi objek dan juga bisa digunakan untuk menghitung jumlah objek yang melaluinya. Spesifikasi dari sensor ini dijelaskan pada Tabel 2.5 dibawah ini.



**Tabel 2.5** Spesifikasi Sensor Infrared Proximity E18-D80NK

Power Supply	5 VDC
supply current	DC 25mA
maximum load current	100mA
response time	2ms
Pointing angle	15 degrees
Detection Range	3-80CM Adjustable
detection of objects	transparent or opaque
working temperature	-25 to +55 degrees C

### 2.6.5 Aktuator

Aktuator adalah sebuah peralatan mekanis untuk menggerakkan atau mengontrol sebuah mekanisme atau sistem. Aktuator diaktifkan dengan menggunakan lengan mekanis yang biasanya digerakkan oleh motor listrik, yang dikendalikan oleh media pengontrol otomatis yang terprogram di antaranya mikrokontroller.

#### 2.6.5.1 Kunci Pintu Elektrik

Kunci pintu elektrik adalah suatu alat mekanis yang bisa digunakan mengunci pintu secara otomatis hanya dengan memberikan tegangan pada kutubnya. *Doorlock* bekerja berdasarkan prinsip solenoid yaitu elektromagnetik yang akan menggerakkan mekaniknya. Gambar 2.12 berikut ini menunjukkan bentuk dari elektrik *Doorlock* tipe LY-03 yang digunakan pada tugas akhir ini.



**Gambar 2.12** Kunci Pintu Elektrik LY-03

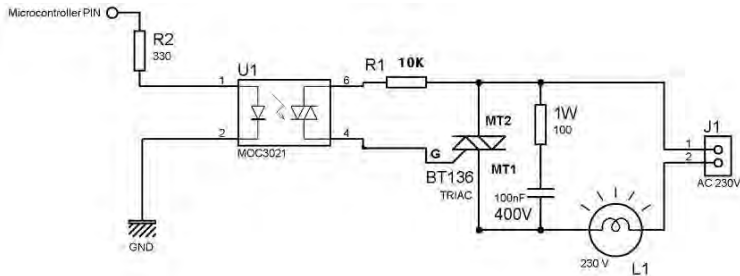
Modul kunci pintu elektrik yang dipakai adalah modul tipe LY-03 dengan spesifikasi lengkap pada Tabel 2.6 berikut:

**Tabel 2.6** Spesifikasi Kunci Pintu Elektrik

Voltage	12VDC
Current	0.35A
Dimension	54*41mm (L*W)
Latch telescopic length	10mm
Energized forms	intermittent
Unlocking time	1S
continuously energized	10S

#### 2.6.5.2 *Driver Lampu*

Driver lampu adalah sebuah rangkaian yang berfungsi untuk mengendalikan nyala lampu, driver dibutuhkan ketika kita ingin mengendalikan suatu perangkat yang batas tegangannya jauh diluar batas dari tegangan operasi dari kontroller yang digunakan, driver juga berfungsi mengisolasi dari tegangan tinggi sehingga mencegah terjadinya kelebihan tegangan. Gambar 2.13 berikut ini menunjukkan skematik rangkaian pengendali lampu.



**Gambar 2.13** Rangkaian pengendali lampu

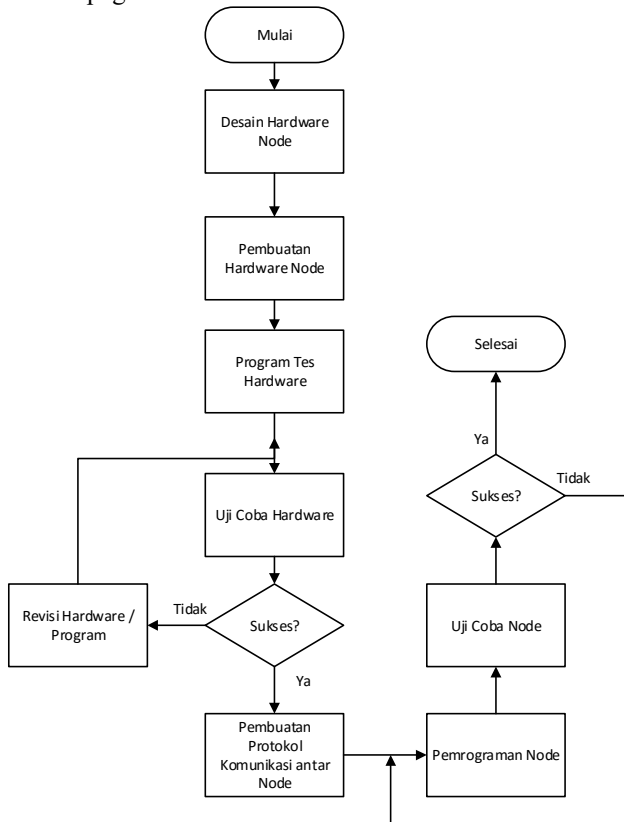
## BAB 3

### PERANCANGAN SISTEM KANTOR PINTAR

#### 3.1 Diagram Alir Perancangan

Tujuan dari tugas akhir ini yaitu merancang dan membuat sistem kantor pintar berbasis *IoT*. Hal ini dilakukan dengan membuat perangkat keras dan perangkat lunak yang saling terintegrasi satu sama lain sehingga membentuk satu kesatuan sistem.

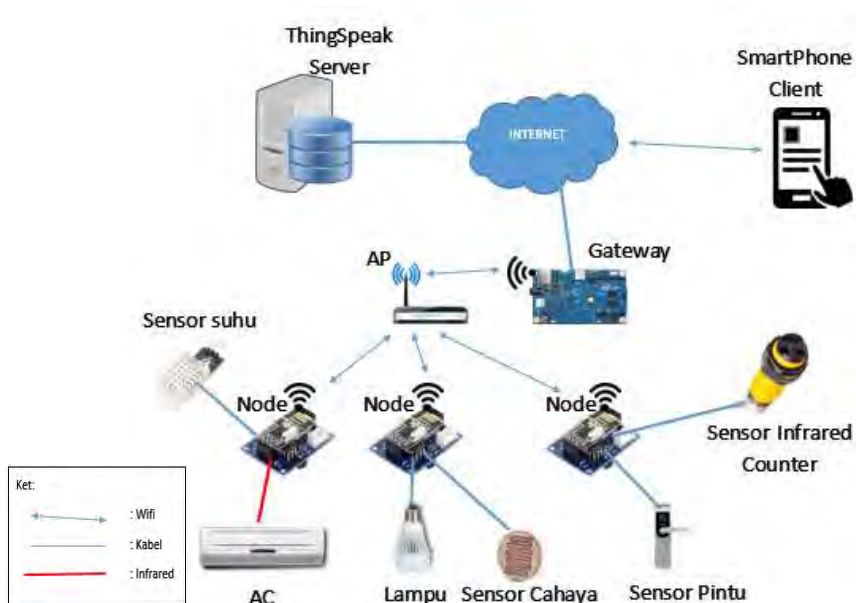
Skema rencana tahapan perancangan digambarkan melalui diagram alir pada Gambar 3.1 berikut.



**Gambar 3.1** Diagram Alir Perancangan sistem

### 3.2 Rancangan Sistem

Gambar 3.2 berikut ini adalah blok diagram sistem secara keseluruhan yang dibuat:



**Gambar 3.2** Blok diagram sistem

Sistem yang akan dibuat terdiri dari 3 buah node sensor, setiap node memiliki fungsionalitas yang berbeda-beda. Setiap node sensor terhubung dengan node gateway sebagai pusat pengendali dari semua node sensor. Tiga buah node sensor yang akan dibuat adalah:

1. Node Sensor dan Kontroller Temperatur Ruangan  
Node ini berfungsi untuk membaca data kondisi temperatur ruangan dimana node ini dipasang, serta node ini bertugas mengirimkan perintah ke sistem pendingin ruangan melalui media komunikasi *infrared*.
2. Node Sensor dan Kontroller Pencahayaan Ruangan

Node ini berfungsi sebagai pengendali nyala dan matinya lampu serta mempunyai fungsi untuk melakukan pembacaan data intensitas cahaya di dalam ruangan berdasarkan sensor yang terpasang.

3. Node sensor dan Kontroller Kunci Pintu

Node ini berfungsi sebagai pengendali kunci pintu, dimana node ini bisa melakukan pembukaan dan penguncian pintu hanya dengan mengatur tegangan yang masuk ke aktuator kunci elektrik, selain itu node ini juga dilengkapi dengan 2 buah sensor inframerah untuk mendeteksi jumlah orang yang berada di dalam ruangan.

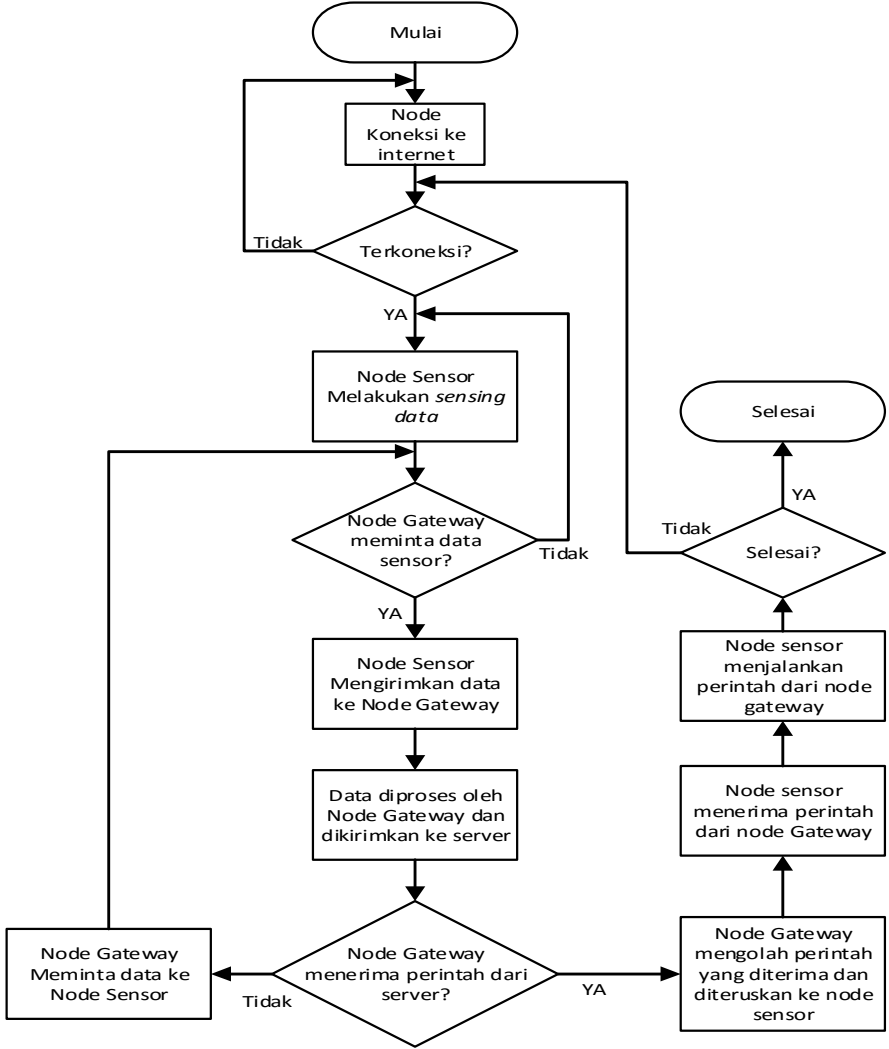
Selain tiga buah node sensor, dibuat juga sebuah node yang berfungsi sebagai gateway ke jaringan internet, node ini menangani semua pertukaran data dari server dengan node sensor.

Sistem ini bekerja mulai dari setiap node melakukan pembacaan sensor kemudian data hasil pembacaan dikirimkan ke node gateway untuk kemudian diteruskan oleh node gateway ke server, kemudian data yang telah ditampung di dalam server diakses oleh aplikasi client android dan ditampilkan dalam bentuk data dan grafik. Pengiriman data sensor ke server harus memiliki interval lebih dari 15 detik, karena sistem pada server membatasi pengiriman data agar tidak terjadi flooding data pada server.

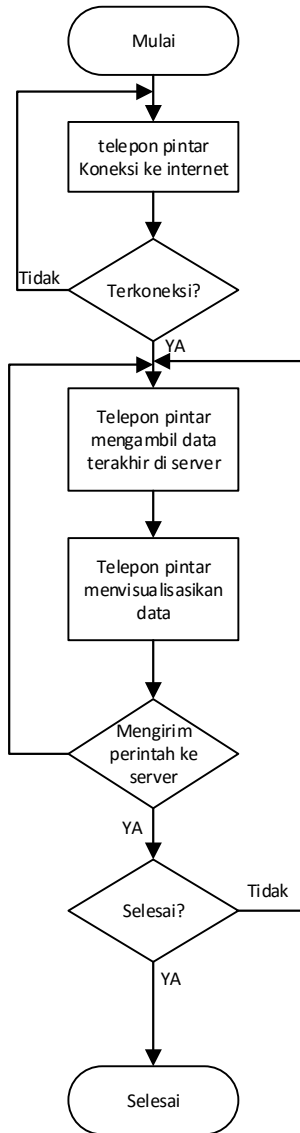
Pengiriman perintah ke node yang bersangkutan dilakukan dengan alur kebalikan dari alur diatas dimana aplikasi client android mula-mula mengirimkan perintah ke server kemudian node gateway membaca perintah tersebut dari server dan meneruskannya ke node yang bersangkutan.

Untuk lebih memperjelas alur kerja dari sistem ini bisa dilihat seperti ditunjukkan pada Gambar 3.3 dan Gambar 3.4.

Diagram alir dari cara kerja sistem ditunjukkan oleh Gambar 3.3 dan Gambar 3.4 berikut ini:



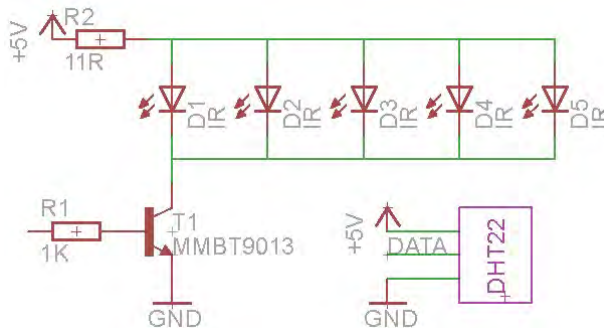
**Gambar 3.3** Diagram Alir Node



**Gambar 3.4** Diagram Alir Aplikasi Telepon Pintar

### 3.2.1 Rancangan Node Kontroller Temperatur

Node kontroller temperatur terdiri dari sebuah *wireless module* ESP8266 (NodeMCU) yang dilengkapi dengan sensor Temperatur beserta 5 buah pemancar *infrared* untuk mengirimkan data ke AC. Node ini bekerja dengan cara membaca data sensor temperatur kemudian data disimpan ke dalam *memory*, ketika node gateway mengirimkan permintaan data maka node ini akan merespon dengan mengirimkan data-data yang diminta oleh node gateway. Rangkaian dari kontroller temperatur ditunjukkan oleh Gambar 3.5. dibawah ini:



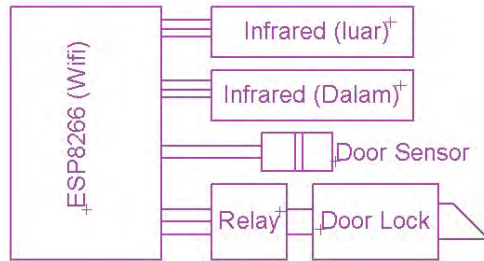
**Gambar 3.5** Rangkaian Kontroller Temperatur

### 3.2.2 Rancangan Node Kontroller Pintu dan Sensor *Visitor Counter*

Node kontroller pintu terdiri dari sebuah *wireless module* ESP8266 yang terintegrasi dengan aktuator *Electric Door Lock* dan sebuah sensor *Magnetic Door Sensor* untuk mengetahui kondisi pintu sedang terbuka atau tertutup, sensor ini bekerja layaknya saklar pada umumnya namun dengan trigger berupa magnet, selain itu node ini juga dilengkapi dua buah *Infrared proximity sensor* untuk digunakan menghitung jumlah pengunjung ruangan, sensor ini bekerja dengan cara mendeteksi sensor mana yang terlebih dahulu mendeteksi objek, sensor luar ataukah sensor dalam, apabila sensor luar yang mendeteksi objek terlebih dahulu maka jumlah orang ditambah dan sebaliknya jika sensor dalam yang terlebih dahulu mendeteksi objek maka jumlah orang



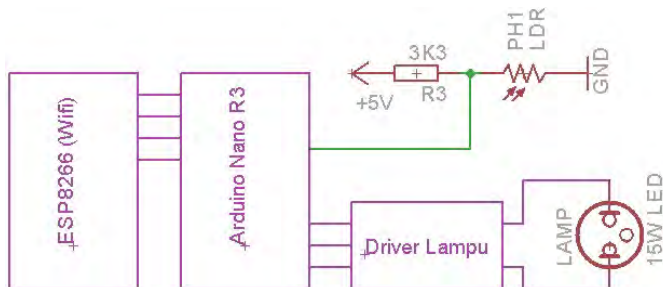
dikurangi. Lebih jelasnya bisa dilihat di diagram blok controller pintu dan sensor visitor counter pada Gambar 3.6 berikut:



**Gambar 3.6** Blok Kontroller Pintu dan Sensor *Visitor Counter*

### 3.2.3 Rancangan Node Kontroller Pencahayaan Ruangan

Node kontroller pencahayaan ruangan terdiri dari sebuah modul wireless ESP8266 yang disambungkan dengan Arduino Nano R3, dimana pada Arduino Nano R3 dipasang 3 buah sensor intensitas cahaya berupa LDR dan 4 buah *Softswitch* sebagai driver untuk mengontrol nyala lampu. Gambar 3.7 dibawah ini menunjukkan blok diagram dari node kontroller pencahayaan ruangan.

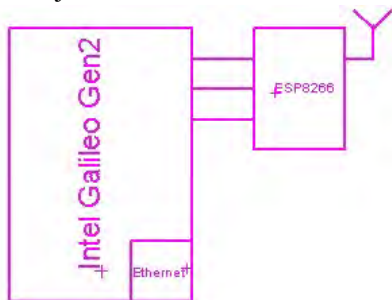


**Gambar 3.7** Blok Node Kontroller Pencahayaan Ruangan

### 3.2.4 Rancangan Node Gateway

Node gateway adalah node kontroller utama dari semua node, jadi semua data dari node akan diproses di node ini dan dikirimkan ke server, untuk itu node ini menggunakan Intel Galileo Gen2 sebagai

kontroller utama namun tidak menutup kemungkinan bisa menggunakan perangkat lain yang memiliki fungsionalitas sama, seperti Arduino, ARM, Raspberry, dll. Pada tugas akhir ini kebetulan penulis menggunakan Intel galileo gen2. *Board* ini menggunakan sistem operasi linux yocto dimana platform ini memiliki fitur bisa menjalankan program berbasis arduino. Untuk komunikasi ke internet node ini menggunakan interface Ethernet sedangkan untuk komunikasi dengan node-node sensor lainnya menggunakan interface wireless berupa wireless modul ESP8266. Diagram bloknya ditunjukkan oleh Gambar 3.8 dibawah ini:

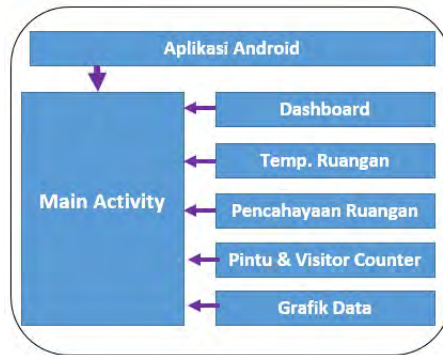


**Gambar 3.8** Blok Node Gateway

### 3.2.5 Rancangan Aplikasi Android

Dalam tugas akhir ini aplikasi android digunakan sebagai antarmuka pengguna dengan sistem *Smart Office*. Aplikasi dibuat dengan pemrograman java khusus untuk android menggunakan *IDE* Android Studio. Di dalam aplikasi dibuat antar muka yang menyajikan semua data-data yang tersimpan di server dan di kemas ke dalam sebuah tampilan agar mudah dipahami oleh pengguna. Dalam aplikasi juga disajikan grafik data sensor yang dibuat menggunakan library “hello chart”. Untuk komunikasi dengan server aplikasi yang dibuat memanfaatkan jaringan internet seluler. Protokol yang digunakan adalah protokol yang bekerja pada layer Aplikasi yaitu protokol HTTP (*Hypertext Transfer Protocol*) dengan menggunakan metode “POST” . metode “POST” diarahkan langsung ke server IoT yang digunakan yaitu “ThingSpeak” dengan menambahkan header data berupa *API Key*.

Blok diagram dari aplikasi android yang digunakan seperti pada Gambar 3.9 berikut ini:



**Gambar 3.9** Rancangan Blok Tampilan Android

Pada Gambar 3.9 aplikasi klient android terdiri dari satu buah *main activity* dan beberapa *fragment*, *fragment – fragment* akan di tampilkan pada *main activity* sesuai dengan menu yang dipilih oleh pengguna.

### 3.3 Implementasi dan Konfigurasi Node

Setelah node jadi, sebelum bisa digunakan perlu dilakukan instalasi dan konfigurasi untuk menyesuaikan dengan kondisi dimana node tersebut akan diterapkan. Instalasi dan konfigurasi juga meliputi pemrograman yang digunakan. Di sisi node diprogram dengan menggunakan bahasa pemrograman C++ serta menggunakan *Library* Arduino. Untuk mempermudah pemrograman penulis menggunakan Visual Studio 2015 sebagai IDE, sedangkan di sisi aplikasi android menggunakan Android Studio sebagai IDE.

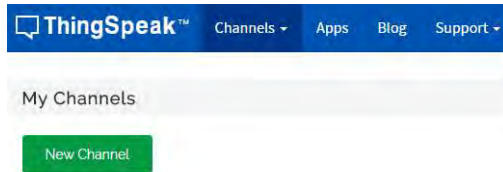
#### 3.3.1 Proses Konfigurasi Server IoT

Server yang digunakan dalam tugas akhir ini memanfaatkan fasilitas yang disediakan oleh MATHWORK berupa server IoT yang diberi nama “ThingSpeak”, server ini bisa diakses di alamat “<http://thingspeak.com>”. Untuk dapat menggunakan server ini terlebih dahulu harus mendaftarkan akun terlebih dahulu. Proses pendaftaran akun sama halnya pendaftaran akun pada website umumnya. Setelah mendaftarkan akun berikutnya adalah proses pembuatan channel dimana channel inilah yang akan menampung data-data sensor yang diunggah.

### 3.3.1.1 Konfigurasi Channel

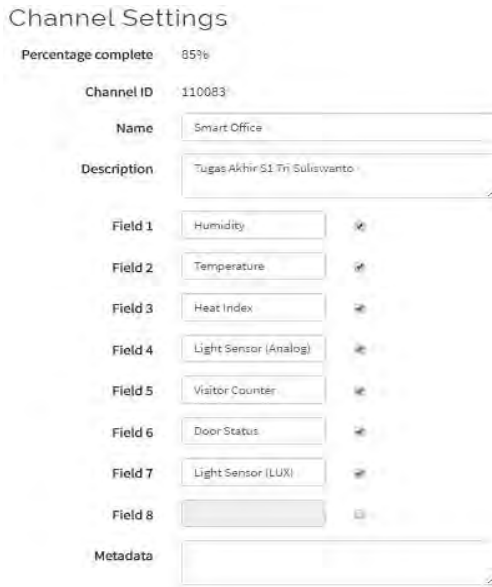
*Channel* merupakan tabel data yang nantinya akan digunakan menyimpan data yang diunggah. Berikut adalah proses pembuatan *channel* dan konfigurasinya:

- a. Pada menu “My Channel” klik “New Channel” seperti pada Gambar 3.10 berikut ini:



**Gambar 3.10** Membuat *Channel*

- b. Kemudian akan dihadapkan dengan “channel setting”, pada tahapan ini perlu diisi field apa saja yang akan digunakan, misalnya seperti Gambar 3.11 berikut:



**Gambar 3.11** Konfigurasi *Channel*

- c. Setelah itu klik pada “API Keys”, di bagian ini akan diperoleh *API Key* yang diperlukan agar kita dapat mengakses channel yang berada di server dengan menggunakan device kita. Gambar 3.12 menunjukkan *API Key* yang didapatkan.

#### Write API Key

Key

[Generate New API Key](#)

#### Read API Keys

Key

Note

[Save Note](#) [Delete API Key](#)

**Gambar 3.12** *API Key*

#### 3.3.1.2 *Konfigurasi TalkBack*

*TalkBack* adalah istilah yang digunakan oleh server thingspeak pada layanannya yang memungkinkan untuk mengirimkan perintah ke perangkat kita. Berikut adalah proses pembuatan dan konfigurasinya:

- a. Masuk ke menu “Apps” dan pilih “TalkBack” kemudian klik “New TalkBack” seperti pada Gambar 3.12 berikut:

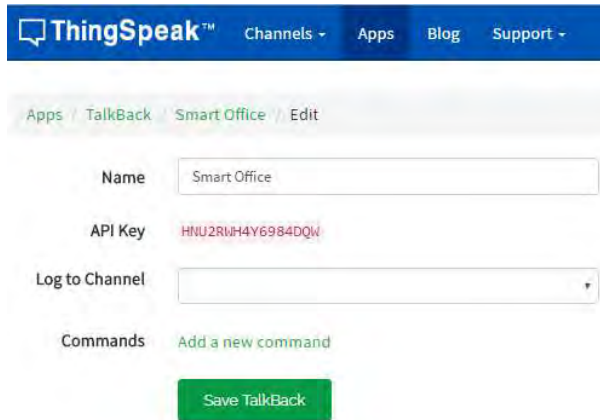
**ThingSpeak™** Channels + Apps Blog Support +

Apps TalkBack

[New TalkBack](#)

**Gambar 3.13** Membuat *TalkBack*

- b. Setelah itu, isi nama sesuai dengan yang diinginkan, pada bagian ini kita akan mendapatkan *API Key* yang harus digunakan untuk dapat mengakses *TalkBack App* seperti yang terlihat di Gambar 3.14 berikut:



ThingSpeak™ Channels ▾ Apps Blog Support ▾

Apps / TalkBack / Smart Office / Edit

Name

API Key HNu2RvH4Y6984DQw

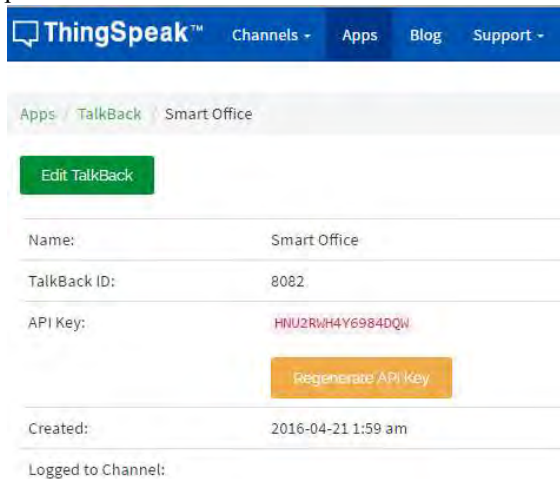
Log to Channel

Commands [Add a new command](#)

[Save TalkBack](#)

**Gambar 3.14** Konfigurasi *TalkBack*

- c. Setelah itu akan mendapatkan *Talkback ID* dan *Talkback API Key* seperti pada Gambar 3.15 berikut:



ThingSpeak™ Channels ▾ Apps Blog Support ▾

Apps / TalkBack / Smart Office

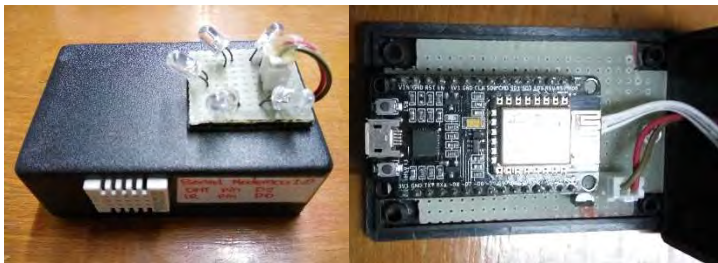
[Edit TalkBack](#)

Name:	Smart Office
TalkBack ID:	8082
API Key:	<span>HNu2RvH4Y6984DQw</span>
	<a href="#">Regenerate API Key</a>
Created:	2016-04-21 1:59 am
Logged to Channel:	

**Gambar 3.15** *TalkBack ID* dan *API Key*

### 3.3.2 Implementasi dan Konfigurasi Node Kontroller Temperature

Setelah perancangan selesai, untuk dapat menggunakan node kontroller temperatur, maka node harus dilakukan instalasi dan konfigurasi terlebih dahulu. Konfigurasi node meliputi dari pemrograman node, konfigurasi pengalamatan IP serta peletakan node. Pada node ini menggunakan kontroller utama ESP8266 sekaligus bertindak sebagai *wireless module* yang diintegrasikan dengan sensor temperatur dan kelembaban ruangan berupa 1 buah sensor DHT22, serta 5 buah pemancar inframerah yang dirangkai secara paralel sebagai media komunikasi antara node dengan sistem pendingin ruangan (AC). Wujud dari node kontroller temperatur diperlihatkan oleh Gambar 3.16 berikut:



**Gambar 3.16** Wujud Node Kontroller Temperature

Node ini bekerja dengan alur membaca data sensor temperatur dan humiditi kemudian ketika node gateway meminta data tersebut maka data tersebut dikirimkan ke node gateway dengan menggunakan protokol komunikasi UDP. Selain itu node ini juga menunggu perintah dari node gateway untuk menaikkan atau menurunkan suhu ruangan dengan cara mengirimkan perintah ke sistem pendingin ruangan. Sebelum bisa mengontrol AC, terlebih dahulu dilakukan pembacaan data protokol data komunikasi yang digunakan oleh AC, itu bisa dilakukan dengan cara menghubungkan infrared receiver ke arduino dan membaca datanya ketika remote AC ditekan tombolnya dan diarahkan ke infrared receiver. Setelah didapatkan data sinyalnya kemudian data sinyal tersebut bisa di bangkitkan ulang dengan memanggil fungsi `irsend.sendLG(data, panjang_bit);` untuk AC merk LG.

Untuk komunikasi antar node dengan node gateway dapat dilakukan dengan beberapa baris potongan program pada Gambar 3.17 berikut:

```

if (packetBuffer[0] == 0x7E && packetBuffer[1] == 0x23) { //cek header ~#
    char Buffer_All[21];
    //Humidity, Temperature, Heat_Index
    strcpy(Buffer_All, "~#"); //kirim header
    strcat(Buffer_All, String(Humidity).c_str());
    strcat(Buffer_All, "#");
    strcat(Buffer_All, String(Temperature).c_str());
    strcat(Buffer_All, "#");
    strcat(Buffer_All, String(Heat_Index).c_str());
    strcat(Buffer_All, "#");
    Response(Buffer_All, UDP.remoteIP(), UDP.remotePort());
}
else if (packetBuffer[0] == 0x7E && packetBuffer[1] == 0x24) { //cek header ~$
    byte Data_Remote = (packetBuffer[2] - 0x30) * 10 + (packetBuffer[3] - 0x30);
    Serial.println(Data_Remote);
    if (Data_Remote == 1){
        irsend.sendLG(LG_Remote[13], 28); //power on
    }
    else if (Data_Remote == 0){
        irsend.sendLG(LG_Remote[14], 28); //power off
    }
    else{
        irsend.sendLG(LG_Remote[Data_Remote - 18], 28);
    }
    Response("OK", UDP.remoteIP(), UDP.remotePort());
}
else {
    Response("Invalid Request!", UDP.remoteIP(), UDP.remotePort());
}

```

**Gambar 3.17** Potongan Program Node Kontroller Temperature

### 3.3.3 Implementasi dan Konfigurasi Node Kontroller Pencahayaan Ruangan

Node kontroller pencahayaan ruangan memiliki 3 buah sensor LDR yang masing-masing dihubungkan ke ADC channel dari Arduino Nano R3, dengan menggunakan fitur ADC dapat dibaca nilai sensor cahaya berupa besaran tegangan yang merepresentasikan intensitas cahaya. Selain sensor cahaya juga dipasang 4 buah pengendali lampu AC 220V untuk mengontrol nyala lampu, pengendali lampu ini beroperasi dengan trigger dari pin arduino, dimana jika pin Arduino di set logic 1 maka tegangan AC220V akan mengalir ke lampu dan menyalakan lampu, sebaliknya jika pin Arduino di set logic 0 maka tidak ada tegangan yang mengalir ke lampu sehingga lampu tidak akan menyala. Untuk Trigger pin Arduino dan mengirimkan data sensor cahaya ke node gateway maka Arduino juga dihubungkan ke sebuah modul wifi ESP8266 dengan interface serial ttl seperti terlihat pada Gambar 3.18 berikut:





**Gambar 3.18** Node Kontroller Pencahayaan Ruangan

Sama halnya dengan node kontroller temperatur, pada node kontroller pencahayaan ini menunggu permintaan dari node gateway untuk mengirimkan data sensornya serta menunggu perintah dari node gateway untuk menyalakan dan mematikan lampu. Gambar 3.19 berikut ini adalah potongan program untuk menangani komunikasi dengan node gateway:

```

if (packetBuffer[0] == 0x7E && packetBuffer[1] == 0x23) { //cek header ~#
  Serial.write(0x23); // # send command read adc to arduino
  //while (Serial.available() == 0); //wait until serial data available
  byte data = Serial.read();
  if (data == 0x23) { //get the header data
    Ldr1_Value = Serial.parseInt();
    Ldr2_Value = Serial.parseInt();
    Ldr3_Value = Serial.parseInt();
  }
  char Buffer_All[21];
  strcpy(Buffer_All, "~#"); //kirim header
  strcat(Buffer_All, String(Ldr1_Value).c_str());
  strcat(Buffer_All, "#");
  strcat(Buffer_All, String(Ldr2_Value).c_str());
  strcat(Buffer_All, "#");
  strcat(Buffer_All, String(Ldr3_Value).c_str());
  strcat(Buffer_All, "#");
  Response(Buffer_All, UDP.remoteIP(), UDP.remotePort());
}

else if (packetBuffer[0] == 0x7E && packetBuffer[1] == 0x24) { //header ~$
  LIGHT_CHANNEL[0] = packetBuffer[2] - 0x30;
  LIGHT_CHANNEL[1] = packetBuffer[3] - 0x30;
  LIGHT_CHANNEL[2] = packetBuffer[4] - 0x30;
  LIGHT_CHANNEL[3] = packetBuffer[5] - 0x30;
  //kirim data ke arduino
  for (byte i = 0; i < 4; i++) {
    Serial.write(0x7E);
    Serial.print(i);
    Serial.println(LIGHT_CHANNEL[i]);
  }
}

```

**Gambar 3.19** Potongan program Node Kontroller Pencahayaan Ruangan

### 3.3.4 Implementasi dan Konfigurasi Node Kontroller Pintu dan Sensor *Visitor Counter*

Node kontroller pintu dan sensor *visitor counter* sama halnya dengan node kontroller temperatur ruangan yaitu menggunakan kontroller utama ESP8266 sekaligus berfungsi sebagai modul wireless. Node ini terintegrasi dengan 2 buah sensor infrared proximity yang digunakan untuk menghitung jumlah orang di dalam ruangan, sensor proximity di letakkan di dekat pintu dengan konfigurasi sejajar dan mengarah ke arah samping pintu, tujuannya adalah ketika ada orang yang keluar masuk melalui pintu maka sensor proximity tersebut bisa mendeteksinya dan menghitung jumlah orang yang berada di dalam ruangan. Selain itu node ini juga mempunyai *Magnetic Door Sensor* yang berfungsi mendeteksi kondisi pintu sedang terbuka atau tertutup. Sensor *Magnetic Door Sensor* diletakkan dengan cara menempelkannya pada daun pintu dan kusen pintu, jika antara sensor yang terdapat di daun pintu dan yang terdapat pada kusen saling berhimpitan maka status pintu dalam kondisi tertutup dan begitu pula sebaliknya. Gambar 3.20 menunjukkan bentuk implementasi dari node.



**Gambar 3.20** Node Kontroller Pintu dan *Visitor Counter*

Informasi hasil pembacaan sensor akan dikirimkan ke node gateway ketika node gateway meminta data tersebut untuk diolah dan ditentukan aksi selanjutnya. Gambar 3.21 berikut adalah potongan program untuk melakukan perhitungan jumlah orang:

```

if (Sensor_Luar == true) {
    Sensor_Luar = false;
    if (State == IDLE) {
        State = MASUK;
    }
    else if (State == KELUAR) {
        if (Jumlah_Orang == 0) Jumlah_Orang = 0;
        else Jumlah_Orang--;
        State = IDLE;
    }
}
if (Sensor_Dalam == true) {
    Sensor_Dalam = false;
    if (State == IDLE) {
        State = KELUAR;
    }
    else if (State == MASUK) {
        if (Jumlah_Orang == 65535) Jumlah_Orang = 65535;
        else Jumlah_Orang++;
        State = IDLE;
    }
}
}

```

**Gambar 3.21** Potongan Program Untuk Menghitung Jumlah Orang

### 3.3.5 Implementasi dan Konfigurasi Node Gateway

Node *gateway* dalam tugas akhir ini adalah node utama yang memegang kendali atas semua node yang ada, mulai dari menerima semua data sensor dari semua node dan mengolahnya serta sebagai gerbang untuk menghubungkan sistem ke jaringan internet dan mengunggah data ke server IoT serta melakukan pengecekan data perintah dari server. Node *gateway* dibangun dengan controller utama sebuah *mini PC* Intel Galileo Gen2. Intel Galileo Gen2 menggunakan linux Yocto sebagai sistem operasinya. Intel Galileo Gen2 memiliki keunggulan dibanding mini PC jenis lainnya karena dengan Intel Galileo Gen2 kita bisa menjalankan program yang dibuat dengan *library* Arduino sehingga mempermudah proses pemrograman. Program yang dibuat di node ini meliputi program untuk meminta data sensor ke masing-masing node beserta algoritma pemisahan data, kemudian program untuk pengolahan data sensor yang diterima untuk kemudian diunggah ke

internet. Selain itu bagian tersulit dari pengerjaan node ini adalah sinkronisasi komunikasi data dengan masing-masing node, karena keterbatasan sumber daya dari module wireless yang hanya bisa melayani 1 koneksi dalam satu waktu maka perlu dilakukan penjadwalan pengiriman dan penerimaan data ke masing-masing node. Untuk mengunggah data ke Internet digunakan protokol HTTP dengan metode POST, penggunaan protokol ini adalah suatu keharusan karena *server IoT Thingspeak* hanya bisa melayani dengan protokol komunikasi HTTP. Gambar 3.22 berikut adalah potongan program yang dibuat untuk node:

```
void updateThingSpeak(String tsData)
{
    if (client.connect(thingSpeakAddress, 80))
    {
        Serial.println("Connected to ThingSpeak...");
        Serial.println();

        client.print("POST /update HTTP/1.1\n");
        client.print("Host: api.thingspeak.com\n");
        client.print("Connection: close\n");
        client.print("X-THINGSPEAKAPIKEY: " + writeAPIKey + "\n");
        client.print("Content-Type: application/x-www-form-urlencoded\n");
        client.print("Content-Length: ");
        client.print(tsData.length());
        client.print("\n\n");

        client.print(tsData);

        lastConnectionTime = millis();

        resetCounter = 0;
    }
    else
    {
        Serial.println("Connection Failed.");
        Serial.println();

        resetCounter++;

        if (resetCounter >= 5) { startEthernet(); }

        lastConnectionTime = millis();
    }
}
```

**Gambar 3.22** Potongan Program Untuk Mengunggah Data ke Server

Intel Galileo Gen2 dalam tugas akhir ini diintegrasikan dengan sebuah modul wireless ESP8266 sebagai antarmuka untuk komunikasi ke node-node lainnya. Gambar 3.23 berikut adalah bentuk dan konfigurasi dari Node gateway yang digunakan.



**Gambar 3.23** Node Gateway

Komunikasi antara Intel Galileo dengan modul wireless menggunakan antar muka serial ttl dengan baudrate 11500, sedangkan untuk komunikasi node *gateway* ini dengan server menggunakan antarmuka sebuah ethernet port.

### **3.3.6 Implementasi dan Konfigurasi Aplikasi Android**

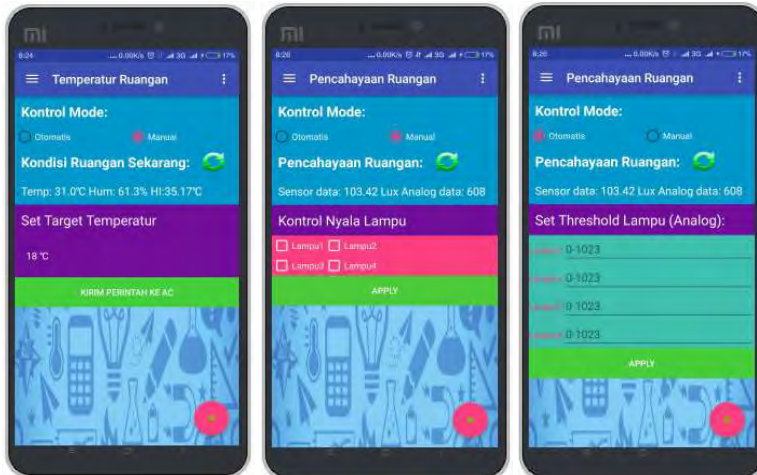
Aplikasi android dalam tugas akhir ini digunakan sebagai antarmuka ke pengguna layanan kantor pintar. Aplikasi ini dibuat dengan tujuan memiliki antarmuka yang sederhana dan dapat dioperasikan dengan mudah. Aplikasi ini dibuat menggunakan Android Studio IDE dan dengan Android SDK sebagai *library* utama. Isi dari aplikasi android yang dibuat mencakup tampilan informasi sensor baik berupa dalam bentuk grafik maupun berupa informasi teks serta berisi antarmuka untuk melakukan pengontrolan node. Gambar 3.24 dan Gambar 2.5 menunjukkan tampilan antarmuka pada aplikasi android yang dibuat.



(a)

(b)

**Gambar 3.24** (a). Tampilan Utama Aplikasi (b). Tampilan Saat Menu di Buka



(a)

(b)

(c)

**Gambar 3.25** Tampilan (a). Temperatur Ruangan (b). Pencahayaayan ruangan Manual (c). Pencahayaayan Ruangan Otomatis

Aplikasi ini mengambil data dari server *IoT* kemudian di visualisasikan. Pengambilan data dari server dilakukan dengan protokol komunikasi HTTP sesuai API yang diberikan oleh server *ThingSpeak*. Untuk melakukannya di pemrograman *Android* dapat dilakukan dengan menggunakan *Thingspeak Android Library*. Gambar 3.26 berikut adalah potongan program untuk mengambil data dari server:

```
start = new Date();
// Fetch the specific Channel feed
tsChannel.loadLastEntryInChannelFeed();
// Connect to ThinkSpeak Channel 110083
tsChannel = new ThingSpeakChannel(110083);
tsChannel.setFeedUpdateListener(new
ThingSpeakChannel.FeedEntryUpdateListener() {
@Override
public void onFeedUpdated(long channelId, long entryId, Feed
feed) {
    double voltage, Lux;
    int Analog, Door_Status;
    String Door_status;
    Analog = (int) Integer.valueOf(feed.getField4());
    voltage = Analog * (5.0 / 1023.0);
    Lux = ((2500/voltage)-500)/3.3;
    Door_Status = (int)
Integer.valueOf(feed.getField6());
    if (Door_Status == 1) {
        Door_status = "PINTU TERBUKA";
    } else {
        Door_status = "PINTU TERTUTUP";
    }
    Loading_panel.setVisibility(View.GONE);
    end = new Date();
    String text = "Elapsed:" + (end.getTime() -
start.getTime()) + "ms";
    Toast.makeText(getActivity(), text,
Toast.LENGTH_LONG).show();
}
});
```

**Gambar 3.26** Potongan Program Pengambilan Data dari Server

### 3.4 Uji Coba Sistem

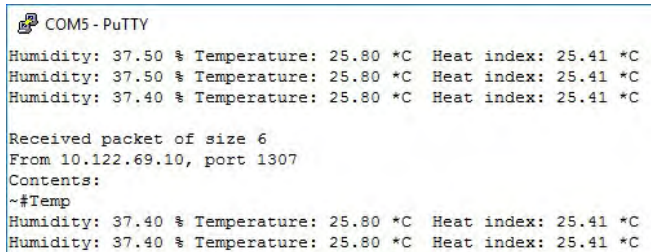
Setelah sistem yang dibuat telah jadi sepenuhnya, maka langkah berikutnya adalah melakukan uji coba fungsionalitas sistem yang telah dibuat.



### 3.4.1 Uji Coba Node Kontroller Temperatur

Uji coba pengontrol temperatur meliputi pengetesan fungsionalitas sensor temperatur serta uji coba kemampuan wireless modul untuk mengirimkan atau menerima data.

Uji coba yang dilakukan adalah pembacaan data sensor temperatur dan pengiriman data ke node gateway. Berikut adalah gambar hasil uji coba, Gambar 3.27 berikut diambil dari sisi node gateway sehingga sudah memastikan bahwa komunikasi antar node sukses.



```
COM5 - PuTTY
Humidity: 37.50 % Temperature: 25.80 *C Heat index: 25.41 *C
Humidity: 37.50 % Temperature: 25.80 *C Heat index: 25.41 *C
Humidity: 37.40 % Temperature: 25.80 *C Heat index: 25.41 *C

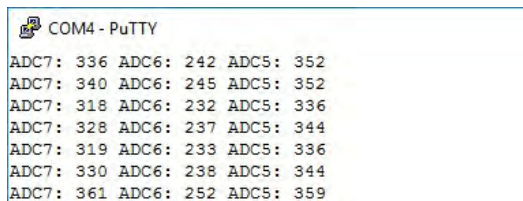
Received packet of size 6
From 10.122.69.10, port 1307
Contents:
~#Temp
Humidity: 37.40 % Temperature: 25.80 *C Heat index: 25.41 *C
Humidity: 37.40 % Temperature: 25.80 *C Heat index: 25.41 *C
```

**Gambar 3.27** Uji Coba Node Kontroller Temperatur

### 3.4.2 Uji Coba Node Kontroller Pencahayaan Ruang

Uji coba node kontroller pencahayaan meliputi pengetesan fungsionalitas sensor cahaya (LDR) dan fungsionalitas pengendali lampu serta pengetesan komunikasi dengan node gateway.

Uji coba yang dilakukan adalah pembacaan sensor cahaya dan pengiriman atau penerimaan data. Gambar 3.28 Berikut adalah gambar hasil uji coba, gambar berikut diambil dari sisi node gateway sehingga sudah memastikan bahwa komunikasi antar node sukses.



```
COM4 - PuTTY
ADC7: 336 ADC6: 242 ADC5: 352
ADC7: 340 ADC6: 245 ADC5: 352
ADC7: 318 ADC6: 232 ADC5: 336
ADC7: 328 ADC6: 237 ADC5: 344
ADC7: 319 ADC6: 233 ADC5: 336
ADC7: 330 ADC6: 238 ADC5: 344
ADC7: 361 ADC6: 252 ADC5: 359
```

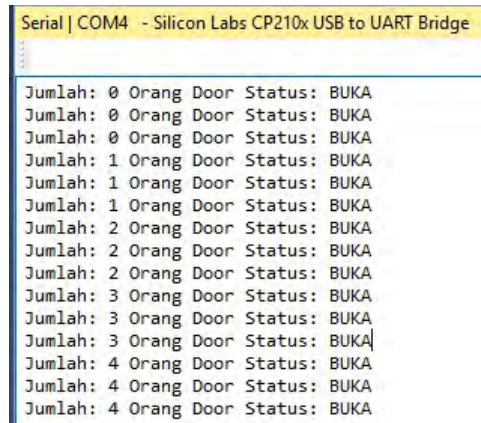
**Gambar 3.28** Uji Coba Node Kontroller Pencahayaan Ruang



### 3.4.3 Uji Coba Node Kontroller Pintu dan Sensor *Visitor Counter*

Uji coba node kontroler pintu dan sensor *Visitor Counter* meliputi pengetesan fungsionalitas sensor *Visitor Counter* dan fungsionalitas pengendali kunci pintu serta pengetesan komunikasi dengan node gateway.

Uji coba yang dilakukan adalah pembacaan sensor *Visitor Counter* serta sensor pintu dan pengiriman data ke node gateway. Berikut adalah gambar hasil uji coba, Gambar 3.29 berikut diambil dari sisi node gateway sehingga sudah memastikan bahwa komunikasi antar node sukses.

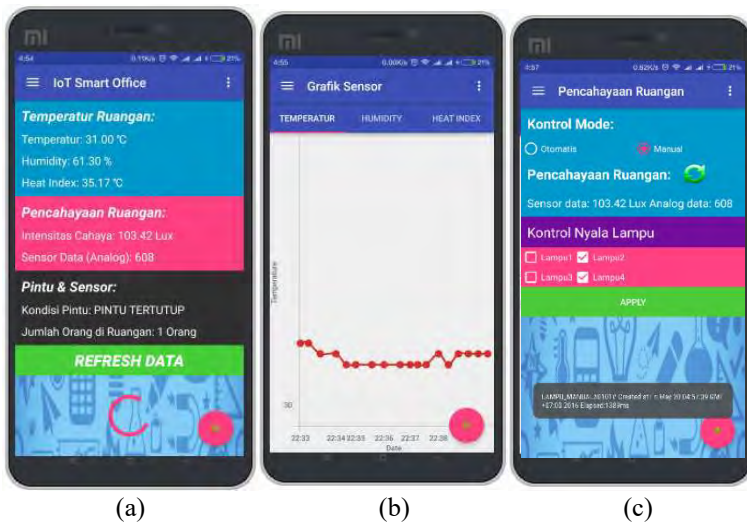


**Gambar 3.29** Uji Coba Node Kontroler Pintu dan Sensor *Visitor Counter*

### 3.4.4 Uji Coba Aplikasi Android

Uji coba aplikasi android dilakukan dengan cara menjalankan aplikasinya serta mencoba berbagai fitur yang ada, seperti fitur mengambil data dari server serta menampilkan data dari server dalam bentuk grafik dan juga melakukan uji coba mengirim perintah ke server.

Gambar 3.30 berikut ini menampilkan proses uji coba fitur yang dimiliki aplikasi android yang telah dibuat.



**Gambar 3.30** Uji Coba (a). Ambil Data dari Server (b). Membuat Grafik Data (c). Kontrol Node

### 3.5 Skenario Pengukuran

Skenario pengukuran dilakukan untuk menentukan bagaimana sistem diukur keandalannya.

#### 3.5.1 Pengukuran Sensor Temperatur Ruangan

Pengukuran sensor temperatur ruangan dilakukan di dalam ruangan dengan membandingkan data dari sensor temperatur dengan data yang terbaca pada alat termometer dan pada suhu yang di set pada sistem pendingin ruangan (AC). Pertama-tama node sensor dinyalakan dan dibaca datanya melalui port serial yang tersedia dan saat bersamaan dibaca juga data dari termometer sebagai pembanding. Pengukuran dilakukan dengan memvariasikan parameter suhu pada sistem pendingin ruangan (AC). Gambar 3.31 berikut ini menampilkan skenario pengukuran sensor temperatur ruangan.



**Gambar 3.31** Skenario Pengukuran Sensor Temperatur Ruang

### 3.5.2 Pengukuran Sensor Intensitas Cahaya

Pengukuran sensor intensitas cahaya yaitu LDR dilakukan dengan metode memvariasikan kondisi pencahayaan di tempat di mana sensor digunakan dengan tujuan untuk mendapatkan data hubungan antara nilai sensor dengan besarnya intensitas cahaya pada ruangan tersebut. Skenario pengukuran sensor intensitas cahaya ditunjukkan oleh Gambar 3.32 berikut ini:



**Gambar 3.32** Skenario Pengukuran Sensor Intensitas Cahaya

Variasi kondisi pencahayaan diperoleh dengan cara mengubah jarak sumber cahaya dengan sensor cahaya.

### 3.5.3 Pengukuran Sensor *Visitor Counter*

Skenario untuk mengukur keakurasian sensor *visitor counter* dalam menghitung jumlah orang di dalam ruangan yaitu dengan cara

menempatkan sensor di muka pintu. pengukuran dilakukan dengan mencoba masuk ke dalam ruangan dan keluar ruangan. Keandalan sensor akan ditentukan berdasarkan keakurasian sensor dalam menghitung jumlah orang. Skenario pengukuran sensor visitor counter ditunjukkan oleh Gambar 3.33 berikut.



**Gambar 3.33** Skenario Pengukuran Sensor *Visitor Counter*

Sensor infrared diletakkan pada posisi 1 meter dari atas lantai, tujuannya adalah bagian tubuh yang terdeteksi adalah bagian pinggang keatas.

#### **3.5.4 Pengukuran Keandalan Kontroller Temperatur**

Untuk menguji keandalan node kontroller temperatur ruangan dilakukan dengan skenario node diletakkan dengan posisi infrared transmitter menghadap *AC* tanpa terhalang apapun. Pengukuran dilakukan dengan 2 mode yaitu mode kontrol otomatis dan kontrol manual. Kontrol manual maksudnya adalah langsung mengatur *AC* pada level suhu tertentu, sedangkan kontrol mode otomatis maksudnya adalah mengatur agar *AC* otomatis melakukan pengaturan levelnya menyesuaikan target dari suhu yang diinginkan.

Pengukuran keandalan mode otomatis dilakukan dengan memilih mode otomatis pada aplikasi android dan menentukan target temperatur yang diinginkan, parameter keandalan ditentukan berdasarkan seberapa lama node kontroller merespon dan mengatur *AC* sampai target level yang diinginkan.

Pengukuran keandalan mode manual dilakukan dengan memilih mode manual dan memilih target level dari AC, parameter keandalan diukur berdasarkan delay waktu yang dibutuhkan node hingga merespon perintah dari aplikasi android.

### **3.5.5 Pengukuran Keandalan Kontroller Pintu**

Pada node kontroller pintu terdapat aktuator berupa kunci pintu elektrik. Kunci pintu elektrik secara normal berada pada kondisi mengunci, untuk membukanya maka perlu dialiri tegangan sebesar 12V DC. Pemberian tegangan ini diatur oleh modul wifi yang digunakan yang terhubung dengan driver.

Skenario pengukuran keandalan kontroller pintu adalah dengan menghitung tingkat keberhasilan kunci pintu elektrik merespon perintah yang dikirimkan serta waktu yang dibutuhkan untuk merespon. Pengiriman perintah dilakukan dengan aplikasi android.

### **3.5.6 Pengukuran Keandalan Kontroller Lampu**

Node kontroller lampu memiliki kemampuan untuk mengatur nyala lampu dengan 2 mode, yaitu mode otomatis dan mode manual. Untuk mengetahui keandalan dari kontroller ini dilakukan pengukuran pada masing-masing mode, yaitu mode otomatis dan mode manual.

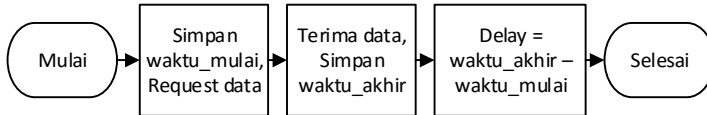
Pengukuran mode otomatis dilakukan dengan mengatur batas dari sensor cahaya di aplikasi android, kemudian parameter yang diukur adalah keberhasilan dari kontroller merespon, dimana besarnya cahaya yang diterima sensor cahaya di atur dengan memberikan cahaya menghadap sensor.

Pengukuran mode manual dilakukan dengan cara memberikan perintah kepada node kontroller untuk mematikan atau menyalakan lampu melalui aplikasi android. Parameter yang diukur pada mode ini adalah respon waktu yang diperlukan kontroller untuk merespon perintah mulai dari saat perintah dikirimkan sampai dengan lampu bisa nyala atau mati.

### **3.5.7 Pengukuran Delay Pengiriman Data Antar Node**

Untuk mengukur delay atau waktu tunda pengiriman data antar node, dalam hal ini adalah antar node gateway dengan masing-masing node sensor dilakukan penambahan program *timer* di sisi node gateway untuk menghitung waktu tunda dari proses pengiriman data antar node.

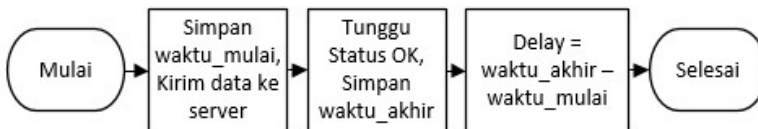
Skenarionya adalah node gateway mengirimkan perintah ke node sensor yang mengindikasikan node gateway meminta data sensor untuk dikirimkan, waktu tunda inilah yang diukur dalam pengukuran ini seperti blok diagram pada Gambar 3.34 berikut.



**Gambar 3.34** Skenario Pengukuran Delay Pengiriman Data antar Node

### 3.5.8 Pengukuran Delay Pengiriman Data ke Server IoT

Untuk mengukur delay atau waktu tunda pengiriman data sensor ke server IoT dilakukan dengan menggunakan skenario seperti pada Gambar 3.35 berikut:



**Gambar 3.35** Skenario Pengukuran Delay Pengiriman Data ke Server

Pengukuran dilakukan dengan menggunakan acuan waktu sesaat sebelum node gateway mengirimkan data ke server dengan protokol HTTP sebagai waktu awal. Setelah data terkirim ke server maka server akan memberikan *ack* berupa kode status “OK” yang mengindikasikan pengiriman sukses, kemudian waktu akhir juga disimpan. Delay waktu pengiriman dihitung dari waktu akhir dikurangi dengan waktu awal.

### 3.5.9 Pengukuran Delay Pengiriman Perintah Dari Android ke Server IoT

Skenario untuk mengukur delay pengiriman perintah dari aplikasi android ke server adalah dengan menambahkan program pembaca waktu ketika *event* pengiriman terjadi dan ketika terjadi *event* respon *ack* yang mengindikasikan pengiriman sukses. Selisih waktu antara *event* pengiriman dan *event* respon *ack* dihitung sebagai delay

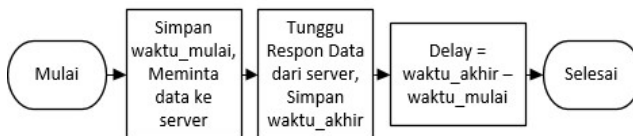
pengiriman. Gambar 3.36 berikut ini adalah blok diagram skenario pengukuran:



**Gambar 3.36** Skenario Pengukuran Delay Pengiriman Data ke Server dari Aplikasi Android

### 3.5.10 Pengukuran Delay Pengambilan Data Server IoT dari Android

Delay proses pengambilan data dari server IoT diukur dengan skenario aplikasi android mengirimkan permintaan data ke server dengan menyertakan *API KEY* yang didapatkan dari proses registrasi sebelumnya, kemudian menunggu respon data dari server. Waktu dari proses inilah yang diukur sebagai delay proses pengambilan data. Blok diagram pada Gambar 3.37 berikut menunjukkan skenario pengukuran delay pengiriman data ke server.

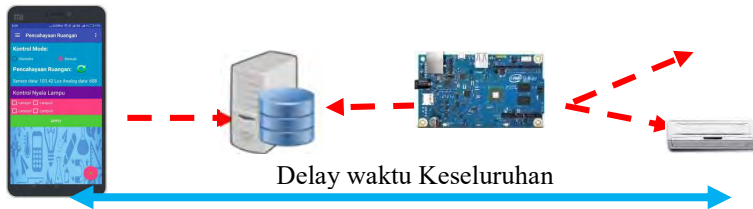


**Gambar 3.37** Skenario Pengukuran Delay Pengiriman Data ke Server

### 3.5.11 Pengukuran Delay Keseluruhan Sistem

Skenario pengukuran delay keseluruhan sistem dilakukan dengan mengukur waktu proses mulai dari aplikasi android mengirimkan perintah ke server hingga node gateway membaca perintah yang terdapat di server dan menjalankan perintah tersebut sampai node sensor merespon perintah tersebut. Pengukuran tidak bisa dilakukan menggunakan fungsi pewaktuan pada perangkat karena perintah yang dikirimkan android tidak langsung dikirimkan ke node melainkan perintah tersebut disimpan di server terlebih dahulu hingga node gateway melakukan sekuen untuk

memeriksa perintah yang ada di server, oleh sebab itu pengukuran waktu dilakukan dengan menggunakan *stopwatch*. Pengukuran dilakukan dengan node terhubung dengan jaringan internet kampus ITS serta aplikasi android menggunakan koneksi jaringan internet kampus ITS dan menggunakan jaringan internet seluler. Gambar 3.38 menunjukkan skenario pengukuran delay keseluruhan sistem.



**Gambar 3.38** Skenario Pengukuran Delay Keseluruhan Sistem



## BAB 4

### ANALISIS KINERJA SISTEM

Pada bab ini akan membahas mengenai analisis dari proses pengukuran di bab 3 yaitu, analisis performansi sistem *Smart Office* berbasis *IoT* yang dibuat.

#### 4.1 Analisis Pengukuran Sensor Temperatur Ruangan

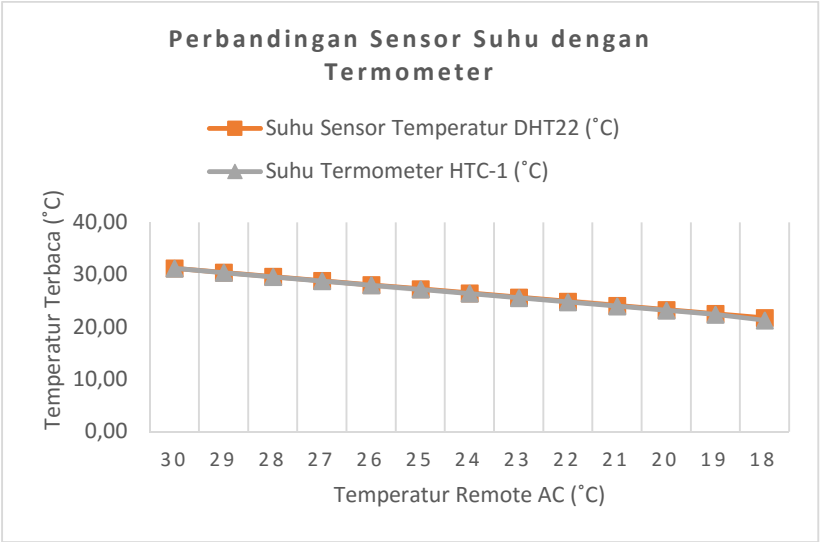
Pengujian ini dilakukan untuk mengetahui tingkat keakurasian sensor dalam melakukan pembacaan data temperatur dibandingkan dengan termometer ruangan konvensional. Tabel 4.1 berikut merupakan data hasil pengukuran sensor temperatur ruangan yang digunakan pada node sensor dalam tugas akhir ini:

**Tabel 4.1** Hasil Pengukuran Sensor Temperatur dan Termometer

No.	Suhu			Error (%)
	AC (°C)	Sensor Temperatur DHT22 (°C)	Termometer HTC-1 (°C)	
1	30	31,24	31,20	0,13
2	29	30,44	30,40	0,13
3	28	29,66	29,60	0,20
4	27	28,87	28,80	0,23
5	26	28,08	28,00	0,27
6	25	27,29	27,20	0,32
7	24	26,50	26,40	0,36
8	23	25,71	25,60	0,41
9	22	24,92	24,80	0,47
10	21	24,13	24,00	0,53
11	20	23,34	23,20	0,59
12	19	22,55	22,40	0,65
13	18	21,76	21,30	2,10

Dari Tabel 4.1 dapat dilihat bahwa dari 13 kali percobaan dengan memvariasikan temperatur pada sistem pendingin ruangan mulai dari diatur pada tingkat suhu 18°C hingga 30°C didapatkan hasil yang sangat akurat dengan error yang kecil, hanya pada saat diatur pada tingkat 18°C pembacaan sensor cenderung lebih tinggi dibandingkan dengan hasil

pembacaan dari termometer. Kondisi error yang terjadi selama pengukuran maksimum hanya 2,10%, dimana error yang terjadi termasuk dalam kondisi wajar. Gambar 4.1 menunjukkan grafik pengukuran sensor suhu dengan termometer HTC-1



**Gambar 4.1** Grafik Pengukuran Sensor Suhu dengan Termometer

Gambar 4.1 di atas menunjukkan kurva perbandingan antara pembacaan sensor temperatur yang digunakan oleh node yaitu sensor DHT22 dengan termometer ruangan tipe HTC-1. Terlihat sekali bahwa data hasil pengukuran selisihnya sangat kecil sehingga di kurva terlihat berhimpit, ini menunjukkan keakurasian yang dimiliki sensor suhu DHT22 sangat akurat terhadap temperatur ruangan HTC-1.

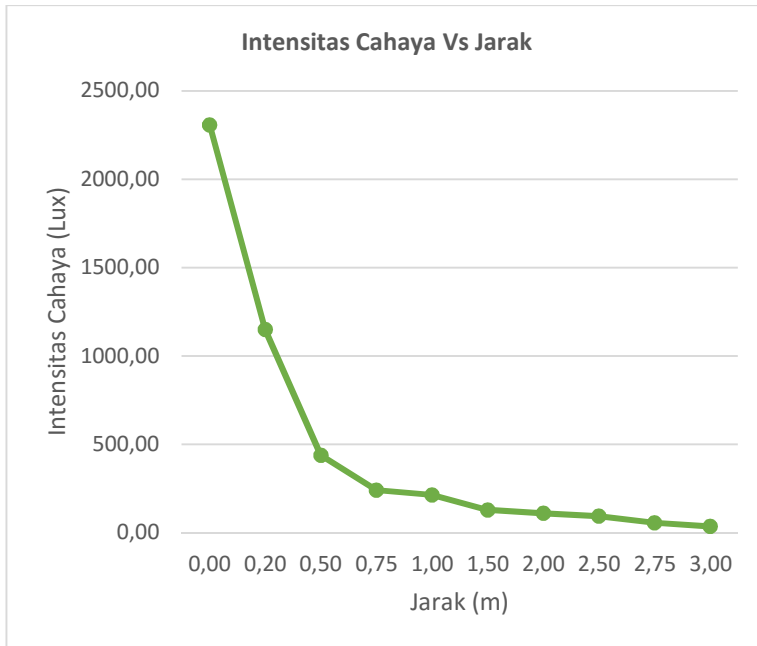
#### 4.2 Analisis Pengukuran Sensor Intensitas Cahaya

Pengukuran sensor intensitas cahaya dilakukan dengan tujuan untuk mengetahui sensitifitas sensor terhadap perubahan intensitas cahaya. Tabel 4.2 berikut merupakan data hasil pengukuran berdasarkan Skenario pada bab 3.5.2:

**Tabel 4.2** Hasil Pengukur Sensor Intensitas Cahaya (LDR)

No.	Jarak (m)	Data Sensor			Rata- rata	Intensitas Cahaya	
		Sensor 1	Sensor 2	Sensor 3			
1	0,00	54	66	69	63,00	2308,80	Lux
2	0,20	124	121	112	119,00	1151,01	Lux
3	0,50	263	260	265	262,67	438,59	Lux
4	0,75	398	395	392	395,00	240,89	Lux
5	1,00	425	422	426	424,33	213,76	Lux
6	1,50	552	547	560	553,00	128,77	Lux
7	2,00	587	591	603	593,67	109,57	Lux
8	2,50	623	642	633	632,67	93,48	Lux
9	2,75	745	747	751	747,67	55,80	Lux
10	3,00	832	829	828	829,67	35,31	Lux

Tabel 4.2 di atas diperoleh dengan melakukan pengukuran menggunakan Sensor cahaya jenis LDR dimana resistansinya akan berubah sesuai intensitas cahaya yang diterima, nilai resistansinya akan semakin kecil ketika menerima intensitas cahaya yang semakin besar, dan sebaliknya resistansinya akan semakin besar jika intensitas cahaya semakin kecil, sehingga data hasil pengujian tersebut cocok sesuai rangkaian pada Gambar 2.10 dimana resistor di hubungkan seri dengan LDR dan diantara resistor dan LDR dibaca tegangannya oleh mikrokontroller. LDR akan berfungsi sebagai pembagi tegangan, jadi ketika LDR resistansinya kecil maka arus akan langsung menuju ground. Sehingga hasil pembacaan data sensor akan kecil juga seperti terlihat pada Gambar 4.2 berikut ini:



**Gambar 4.2** Grafik Pengukuran Intensitas Cahaya Vs Jarak

Dari data pada Tabel 4.2 saat jarak = 0cm nilai sesor cahaya kecil sekali yang berarti kondisi resistansi LDR sangat kecil yang mengindikasikan intensitas cahaya yang diterima sangat besar hingga mencapai 2308,80 Lux. Intensitas cahaya akan semakin berkurang seiring dengan menjauhnya sumber cahaya dari sensor. Untuk memperjelasnya bisa dilihat pada grafik di Gambar 4.2 yang menunjukkan hubungan besarnya intensitas cahaya dengan jarak sumber cahaya.

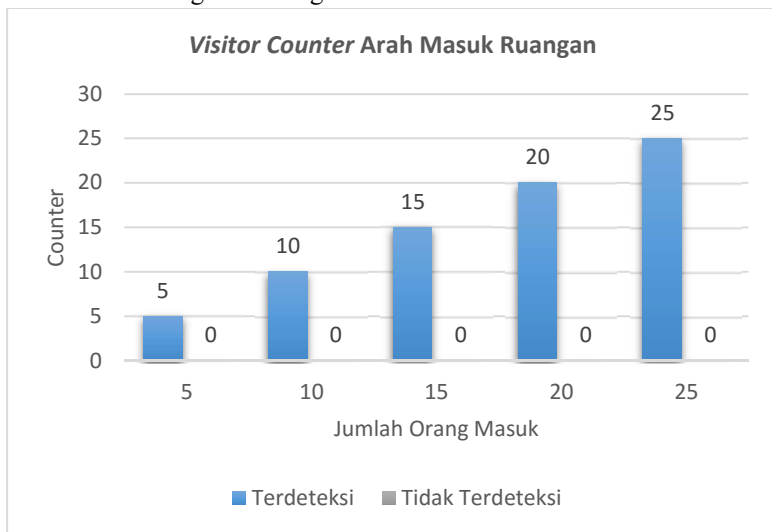
#### 4.3 Analisis Pengukuran Sensor *Visitor Counter*

Pengukuran sensor *Visitor Counter* bertujuan untuk mengetahui seberapa akurat sensor dalam mendeteksi dan menghitung jumlah orang yang berada di dalam ruangan berdasarkan banyaknya orang yang masuk dan yang keluar ruangan. Tabel 4.3 berikut merupakan tabel data hasil pengukuran sensor visitor counter yang dilakukan.

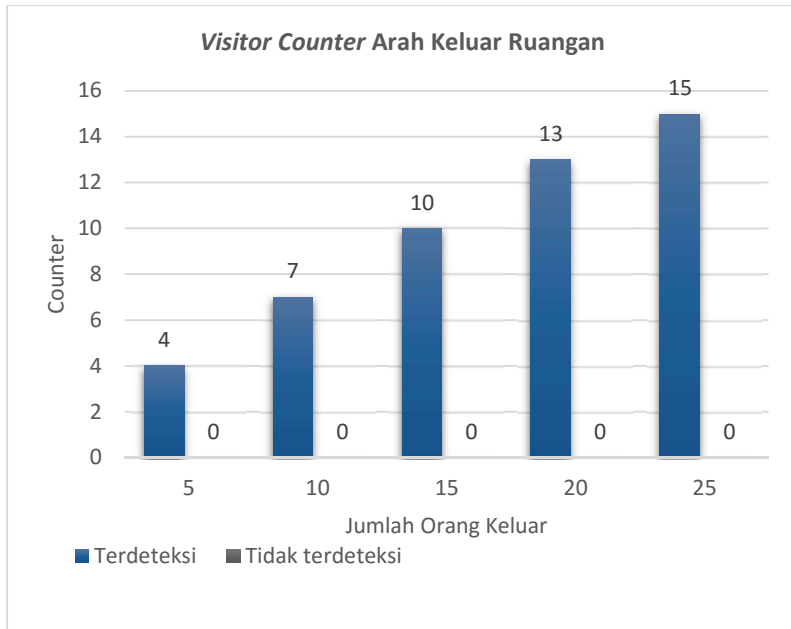
**Tabel 4.3** Hasil Pengukuran Sensor *Visitor Counter*

No	Arah Masuk Ruangan			Arah Keluar Ruangan		
	Jumlah Orang Masuk	Terdeteksi	Tidak Terdeteksi	Jumlah Orang Keluar	Terdeteksi	Tidak terdeteksi
1	5	5	0	4	4	0
2	10	10	0	7	7	0
3	15	15	0	10	10	0
4	20	20	0	13	13	0
5	25	25	0	15	15	0

Dari Tabel 4.3 dapat diketahui dari 5 kali percobaan dengan mencoba keluar masuk ruangan dengan jumlah yang berbeda-beda namun tidak terjadi error pembacaan sama sekali, hal ini menunjukkan bahwa sensor yang digunakan untuk menghitung jumlah orang dalam ruangan sangatlah akurat hingga mencapai tingkat keberhasilan 100%. Untuk lebih jelasnya lihat Gambar 4.3 dan Gambar 4.4, data tersebut disajikan dalam bentuk diagram batang berikut.



**Gambar 4.3** Grafik Pengukuran *Visitor Counter* Arah Masuk Ruangan



**Gambar 4.4** Grafik Pengukuran *Visitor Counter* Arah Keluar Ruangan

#### 4.4 Analisis Keandalan Kontroller Temperatur

Pengukuran keandalan dari kontroller temperatur dilakukan pada 2 mode yang didukung oleh node, yaitu mode otomatis dan mode manual. Tabel 4.4 berikut ini adalah tabel data hasil pengukuran keandalan sistem pada node kontroller temperatur dengan pengaturan mode otomatis serta pengukuran saat belum menggunakan sistem kantor pintar.

**Tabel 4.4** Hasil Pengukuran Keandalan Kontroller Tempertur Mode Otomatis

N o.	Temperat ur Mulai (°C)	Target Temperat ur (°C)	Selisih Temperat ur (°C)	Delay Temperatur Mencapai Target		Selisih Waktu
				Tanpa Sistem	Dengan Sistem	
1	25,90	27	1,1	1 Menit 13 Detik	1 Menit 25 Detik	12 Detik
2	26,13	28	1,87	1 Menit 42 Detik	1 Menit 55 Detik	13 Detik
3	28,14	26	2,14	1 Menit 55 Detik	2 Menit 12 Detik	17 Detik
4	27,46	30	2,54	2 Menit 0 Detik	2 Menit 26 Detik	26 Detik
5	25,96	29	3,04	2 Menit 15 Detik	2 Menit 47 Detik	32 Detik
6	27,04	24	3,04	2 Menit 30 Detik	3 Menit 3 Detik	33 Detik
7	25,42	22	3,42	2 Menit 44 Detik	3 Menit 18 Detik	34 Detik
8	29,04	25	4,04	3 Menit 13 Detik	3 Menit 52 Detik	39 Detik
9	28,15	23	5,15	3 Menit 39 Detik	4 Menit 20 Detik	41 Detik
10	27,19	21	6,19	4 Menit 28 Detik	5 Menit 12 Detik	44 Detik
11	28,77	19	9,77	∞	∞	∞
12	29,05	20	9,05	∞	∞	∞
13	29,49	18	11,49	∞	∞	∞

Pada Tabel 4.4 di atas, pengukuran dilakukan dengan 13 kali percobaan dimana percobaan dilakukan dengan mengatur target temperatur yang diinginkan. Target temperatur yang dimaksud adalah temperatur hasil pembacaan sensor suhu DHT22. Dari data pada Tabel 4.4 diatas dapat diketahui bahwa sistem dapat bekerja dengan optimal ketika target temperatur diset lebih dari 20°C, sedangkan untuk target temperatur kurang dari sama dengan 20°C sistem tidak bisa bekerja, dikarenakan sistem pendingin ruangan yang digunakan untuk pengukuran yaitu AC merek LG dengan kapasitas 1PK tidak mampu untuk mengatur ruangan pada suhu  $\leq 20^{\circ}\text{C}$ , sehingga waktu delay menjadi tak terhingga. Pada percobaan ke 6 dengan kondisi temperatur mulai 27,04°C dan target

temperatur 24°C dengan selisih 3,04°C saat tidak menggunakan sistem membutuhkan waktu sekitar 2 menit 30 detik, sedangkan saat menggunakan sistem kantor pintar waktu yang dibutuhkan sekitar 33 detik lebih lambat dibandingkan dengan menggunakan sistem, hal ini wajar karena sistem menggunakan koneksi internet sehingga ada delay pengiriman perintah ke node.

Tabel 4.5 berikut ini adalah tabel data hasil pengukuran kontroller temperatur pada mode manual. Pengukuran pada mode ini dilakukan dengan mengirimkan perintah berupa level temperatur AC dan bukan berupa nilai suhu yang akan dibandingkan dengan sensor. Pada mode manual ini sistem bekerja dengan open loop, dimana tidak ada umpan balik yang digunakan.

**Tabel 4.5** Hasil Pengukuran Keandalan Kontroller Tempertur Mode Manual

Percobaan	Target Temperatur (°C)	Waktu Respon Kontroller (s)
1	18	7.29
2	19	6.80
3	20	6.26
4	21	5.42
5	22	4.80
6	23	4.17
7	24	3.22
8	25	5.95
9	26	3.65
10	27	5.37
11	28	5.47
12	29	6.77
13	30	4.73
Rata-Rata		5.38

Tabel 4.5 di atas adalah pengukuran pada mode manual dimana dalam mode manual parameter keandalan hanya bisa diukur berdasarkan respon waktu yang diperlukan node untuk mengeksekusi perintah terhitung sejak perintah dikirimkan hingga dieksekusi. Pengujian dilakukan pada AC dengan merek LG dimana memiliki 13 level pengaturan mulai dari 18°C sampai 30°C. Dari data diatas dapat dianalisa



bahwa rata-rata waktu yang dibutuhkan kontroller untuk merespon perintah berkisar 5,38 detik, sedangkan waktu tercepat yang pernah dicapai terjadi pada percobaan ke 7 yaitu 3.22 detik. Waktu dalam pengukuran ini bervariasi tergantung kondisi jaringan saat pengujian.

#### 4.5 Analisis Keandalan Kontroller Pintu

Pengukuran keandalan kontroller pintu diukur berdasarkan kemampuan kontroller dalam memproses perintah yang diberikan. Perintah dikirimkan mulai dari hanya 1 perintah hingga 20 perintah diberikan. Pemberian banyak perintah bertujuan untuk membanjiri sistem dengan perintah dan untuk mengetahui keandalan sistem.

Pada Tabel 4.6 berikut adalah tabel data hasil pengukuran keandalan dari node kontroller pintu.

**Tabel 4.6** Hasil Pengukuran Keandalan Kontroller Pintu

No.	Jumlah Perintah Dikirimkan	Jumlah Perintah Dijalankan	Tingkat Keberhasilan (%)
1	1	1	100
2	2	2	100
3	5	5	100
4	7	7	100
5	9	9	100
6	10	10	100
7	15	15	100
8	20	20	100

Pada Tabel 4.6 pengukuran dilakukan sebanyak 8 kali percobaan dengan memvariasikan jumlah perintah yang diberikan. Dari 8 kali percobaan didapatkan hasil dengan tingkat keberhasilan mencapai 100% yang artinya sistem pada kontroller pintu berjalan dengan sempurna walaupun dilakukan pembanjiran perintah ke sistem.

#### 4.6 Analisis Keandalan Kontroller Lampu

Keandalan dari kontroller lampu diukur pada 2 buah mode yang didukung, yaitu mode kontrol otomatis dan kontrol manual. Pengukuran pertama dilakukan pada mode otomatis dengan 10 kali percobaan. Pada tiap percobaan nilai threshold di atur secara random dengan range 0-1023

(10bit) kemudian pada sensor cahaya diberikan cahaya dengan intensitas tertentu. Parameter keandalan dilihat dari respon yang dihasilkan oleh kontroller dibandingkan dengan respon secara teoritis. Tabel 4.7 berikut adalah data hasil pengukuran keandalan dari kontroller lampu dengan mode otomatis.

**Tabel 4.7** Hasil Pengukuran Keandalan Kontroller Lampu mode otomatis

No.	Threshold Lampu				Data Sensor	Respon Lampu				Respon Secara Teoritis				Presentase Keberhasilan (%)
	1	2	3	4		1	2	3	4	1	2	3	4	
1	472	255	882	831	456	0	1	0	0	0	1	0	0	100
2	988	966	112	576	756	0	0	1	1	0	0	1	1	100
3	172	999	654	443	319	1	0	0	0	1	0	0	0	100
4	766	541	725	204	888	1	1	1	1	1	1	1	1	100
5	638	879	762	128	732	1	0	0	1	1	0	0	1	100
6	903	635	347	931	585	0	0	1	0	0	0	1	0	100
7	600	785	801	757	297	0	0	0	0	0	0	0	0	100
8	789	982	631	629	921	1	0	1	1	1	0	1	1	100
9	982	601	657	886	971	0	1	1	1	0	1	1	1	100
10	278	860	519	221	938	1	1	1	1	1	1	1	1	100
Keterangan: Jika Data Sensor > Threshold Maka Respon Lampu = 1 Selain itu Respon Lampu = 0 0 = Mati 1 = Nyala														

Dari data Tabel 4.7 diatas dapat diketahui dan dianalisa bahwa dari 10 kali percobaan didapatkan presentase keberhasilan 100%, ini menunjukkan bahwa respon kontroller lampu sudah sesuai dengan teori algoritma yang digunakan dimana jika nilai data sensor lebih dari nilai *threshold* yang telah ditetapkan sebelumnya maka lampu akan berlogika 1 atau menyala, dan selain itu lampu akan mati.

Selain mode otomatis, kontroller lampu juga mendukung untuk menggunakan kontrol mode manual, dimana kita bisa leluasa mengatur nyala dan matinya lampu yang kita inginkan dengan hanya mengirimkan perintah melalui aplikasi android yang telah dibuat. Tabel 4.8 berikut ini

adalah tabel hasil pengukuran keandalan dari kontroller lampu pada mode manual.

**Tabel 4.8** Hasil Pengukuran Keandalan Kontroller Lampu Mode Manual

No.	Perintah Lampu				Respon Lampu				Presentase Keberhasilan (%)
	1	2	3	4	1	2	3	4	
1	0	1	1	1	0	1	1	1	100
2	1	1	1	1	1	1	1	1	100
3	0	1	0	1	0	1	0	1	100
4	1	0	0	1	1	0	0	1	100
5	0	0	0	0	0	0	0	0	100
6	1	0	1	1	1	0	1	1	100
7	1	1	1	1	1	1	1	1	100
8	0	0	1	1	0	0	1	1	100
9	0	1	0	0	0	1	0	0	100
10	1	0	0	0	1	0	0	0	100

Tabel 4.8 merupakan pengukuran keandalan pada mode manual ini dilakukan sebanyak 10 kali percobaan dengan memvariasikan perintah yang dikirimkan. Dalam sekali pengiriman perintah langsung dapat melakukan kontrol terhadap 4 buah lampu. Parameter keandalan ditentukan berdasarkan presentase kontroller dapat memproses perintah yang dikirimkan dengan benar. Misal pada percobaan ke 4 perintah yang diberikan adalah Lampu1=1, Lampu2=0, Lampu3=0, Lampu4=1 dan respon yang didapat dari perintah tersebut juga menunjukkan hal yang sama dengan perintah yang dikirimkan yaitu Lampu1=1, Lampu2=0, Lampu3=0, Lampu4=1 sehingga presentase keberhasilannya mencapai 100%.

Dari 10 kali percobaan didapatkan hasil yang sangat memuaskan yaitu berhasil 100% tanpa adanya kesalahan dalam respon yang didapatkan. Hal ini menunjukkan bahwa performansi dari kontroller ini sangatlah baik.

#### 4.7 Analisis Pengukuran Delay Pengiriman Data Antar Node

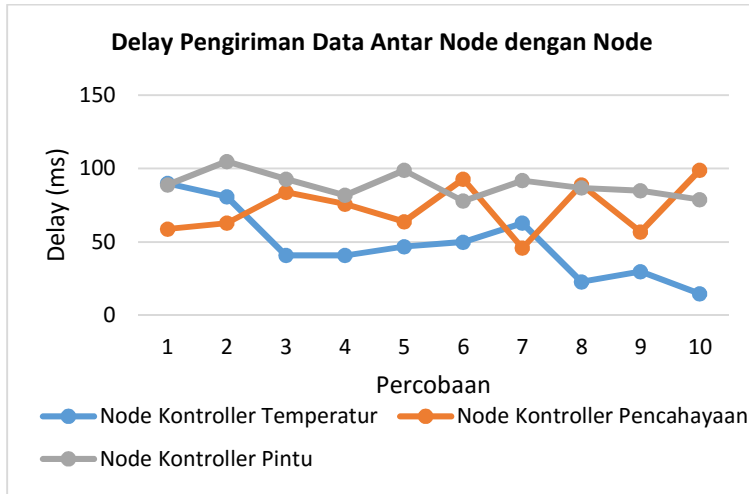
Pengukuran delay pengiriman data antar node bertujuan untuk mengetahui performansi dari node dalam melakukan pengiriman data, dimana data yang ditransmisikan menggunakan protokol UDP pada port 1307. Tabel 4.9 berikut adalah tabel data hasil pengukuran delay komunikasi antar node:

**Tabel 4.9** Hasil Pengukuran Delay Pengiriman Data Antar Node Sensor Dengan Node Gateway

Percobaan	Delay (ms)		
	Node Kontroller Temperatur	Node Kontroller Pencahayaan	Node Kontroller Pintu
1	90	59	89
2	81	63	105
3	41	84	93
4	41	76	82
5	47	64	99
6	50	93	78
7	63	46	92
8	23	89	87
9	30	57	85
10	15	99	79
Rata-Rata	48	73	89

Dari Tabel 4.9 dapat diketahui bahwa delay pengiriman bersifat fluktuatif namun tetap berada pada range tertentu. Untuk delay pengiriman data antar node gateway dengan node kontroller pintu menunjukkan delay paling tinggi di antara node lainnya karena posisi peletakan node ini sedikit lebih jauh dari node gateway serta dipasang didekat pintu ruangan, sehingga delaynya berkisar antara 79ms sampai 105ms. Sedangkan untuk node Kontroller pencahayaan, delay pengiriman berkisar antara 46ms sampai 99ms. Untuk node kontroller temperatur delay pengiriman data hasil pengukuran memiliki delay terendah ini terjadi karena posisi penempatan node lebih dekat dengan node gateway.

Untuk lebih jelasnya data hasil pengukuran delay ketiga node tersebut disajikan dalam bentuk grafik pada Gambar 4.5 berikut ini:



**Gambar 4.5** Grafik Pengukuran Delay Pengiriman Data Antar Node Sensor Dengan Node Gateway

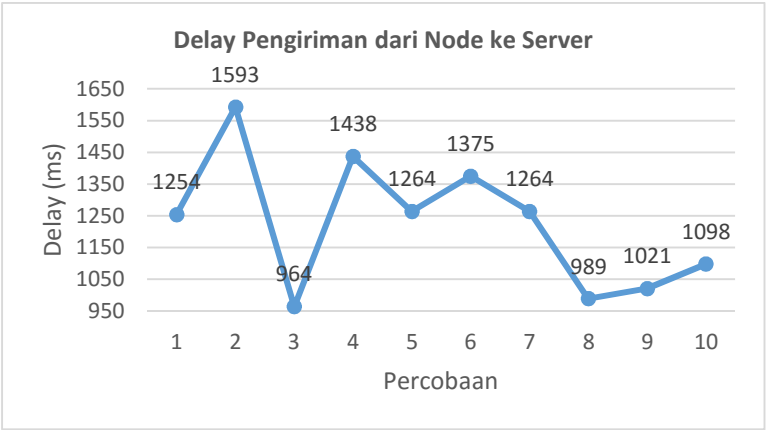
#### 4.8 Analisis Pengukuran Delay Pengiriman Data ke Server IoT

Delay pengiriman data ke server *IoT* diukur untuk mendapatkan performansi node dalam melakukan pengiriman data ke server *IoT* menggunakan jaringan internet. Pengiriman data sensor ke server *IoT* hanya dilakukan oleh node gateway dimana semua data sensor ditampung di node ini baru kemudian dikirimkan ke server *IoT* melalui koneksi jaringan internet. Tabel 4.10 berikut adalah data hasil pengukuran delay pengiriman data ke server *IoT*:

**Tabel 4.10** Hasil Pengukuran Delay Pengiriman Data dari Node Ke Server *IoT*

Percobaan	Delay Pengiriman Data(ms)
1	1254
2	1593
3	964
4	1438
5	1264
6	1375
7	1264
8	989
9	1021
10	1098
Rata-Rata	1226

Pada Tabel 4.10 menunjukkan data hasil pengukuran delay pengiriman data dari node gateway ke server *IoT*. Dari 10 kali percobaan pengukuran didapatkan rata-rata delay pengiriman sebesar 1226 ms. Delay pengiriman bervariasi mulai dari yang terkecil 964 ms hingga delay terlama yaitu 1593 ms. Delay pengukuran ini sangat dipengaruhi oleh kondisi jaringan internet yang digunakan. Untuk memperjelas data hasil pengukuran, pada Gambar 4.6 disajikan data hasil pengukuran dalam bentuk kurva.



**Gambar 4.6** Grafik Pengukuran Delay Pengiriman Data dari Node Ke Server *IoT*

#### 4.9 Analisis Pengukuran Delay Pengiriman Perintah Dari Android ke Server IoT

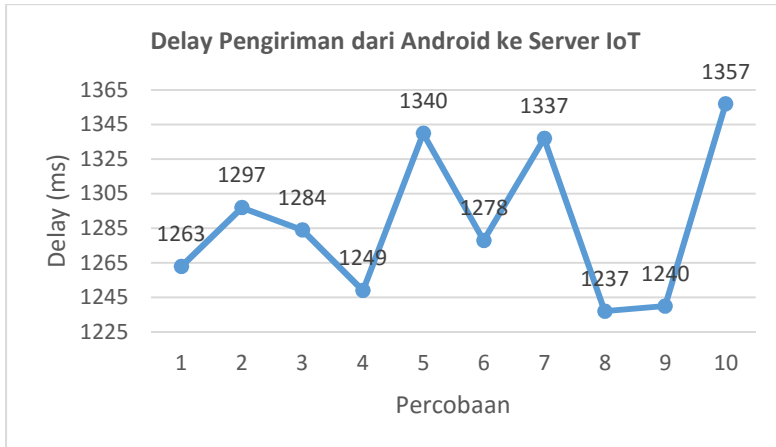
Pengukuran delay pengiriman data dari android ke server *IoT* diukur dengan tujuan mendapatkan lamanya waktu yang dibutuhkan aplikasi android dalam mengirimkan sebuah perintah ke server *IoT*. Pengukuran dilakukan sebanyak 10 kali pengukuran. Pengukuran delay dilakukan sesuai dengan skenario pada bab 3.5.9. delay diukur berdasarkan selang waktu aplikasi melakukan pengiriman data hingga mendapatkan respon berupa pesan ack dari server yang mengindikasikan data yang dikirimkan sukses terkirim dan diterima oleh server. Data hasil pengukuran ditunjukkan pada Tabel 4.11 berikut ini.

**Tabel 4.11** Hasil Pengukuran Delay Pengiriman Data dari Android Ke Server *IoT*

Percobaan	Delay Pengiriman Data(ms)
1	1263
2	1297
3	1284
4	1249
5	1340
6	1278
7	1337
8	1237
9	1240
10	1357
Rata-Rata	1288

Pada Tabel 4.11 dapat dilihat hasil pengukuran delay pengiriman data dari android ke server *IoT* dengan skenario pada bab 3.5.9 berkisar antara 1237 ms sampai dengan 1357 ms, dari 10 kali percobaan pengiriman data dari android ke server *IoT* didapatkan bahwa aplikasi android membutuhkan waktu rata-rata sekitar 1288 ms untuk menyelesaikan proses pengiriman data.

Delay yang didapatkan dari hasil pengukuran ini sangat dipengaruhi oleh kondisi jaringan yang digunakan, dimana dalam pengukuran ini menggunakan jaringan 4G dari operator seluler Indosat Ooredoo. Untuk mempermudah dalam menganalisa data maka data hasil pengukuran ditampilkan dalam bentuk kurva pada Gambar 4.7 berikut ini.



**Gambar 4.7** Grafik Pengukuran Delay Pengiriman Data dari Android ke Server IoT

#### 4.10 Analisis Pengukuran Delay Pengambilan Data Server IoT dari Android

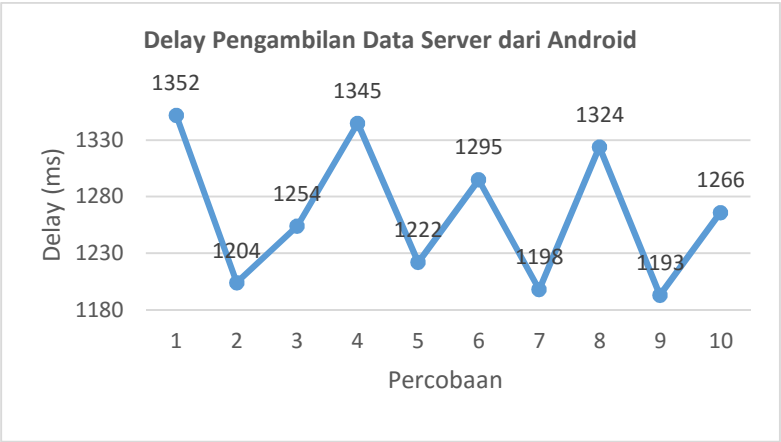
Aplikasi android meminta data ke server dan menunggu sampai data diterima, proses inilah yang diukur sebagai delay pengambilan data server IoT. Penghitungan delay waktu menggunakan fitur library “java.util.Date”. Dengan memanfaatkan library yang ada memungkinkan pengukuran waktu bisa dilakukan secara akurat. Dari Tabel 4.12 terlihat data pengukuran delay pengambilan data server dari aplikasi android yang dilakukan sebanyak 10 kali percobaan berturut-turut. Pengukuran dilakukan sebanyak 10 kali untuk mendapatkan waktu rata-rata yang diperlukan android untuk mengambil data dari server. Tabel 4.12 berikut merupakan data hasil pengukuran delay pengambilan data server *IoT* dari aplikasi android:



**Tabel 4.12** Hasil Pengukuran Delay Pengambilan Data Server IoT dari Android

Percobaan	Delay Pengambilan Data(ms)
1	1352
2	1204
3	1254
4	1345
5	1222
6	1295
7	1198
8	1324
9	1193
10	1266
Rata-Rata	1265

Dari data pengukuran pada Tabel 4.12 juga dapat dihitung rata-rata waktu yang diperlukan untuk melakukan pengambilan data dari server yaitu sekitar 1265 ms. Jarak waktu antara delay terkecil dan delay terbesar hanya sekitar 159 ms. Untuk memperjelas data pada Tabel 4.12 disajikan dalam bentuk grafik pada Gambar 4.8 berikut ini:



**Gambar 4.8** Grafik Pengukuran Delay Pengambilan Data Server IoT dari Android

Dari 10 kali percobaan yang dilakukan, pada Gambar 4.8 terlihat delay waktu yang sifatnya fluktuatif dari nilai terkecil pada percobaan ke 9 yaitu 1193 ms hingga nilai delay terbesar pada percobaan pertama yaitu 1352 ms. Data hasil pengukuran memang fluktuatif namun range fluktuatifnya tergolong kecil.

#### 4.11 Analisis Pengukuran Delay Keseluruhan Sistem

Delay keseluruhan sistem diukur mulai saat aplikasi android mengirimkan perintah ke server hingga node sensor merespon perintah yang dikirimkan. Tabel 4.13 Berikut ini merupakan hasil pengukuran delay keseluruhan sistem sesuai skenario pada bab 3.5.11.

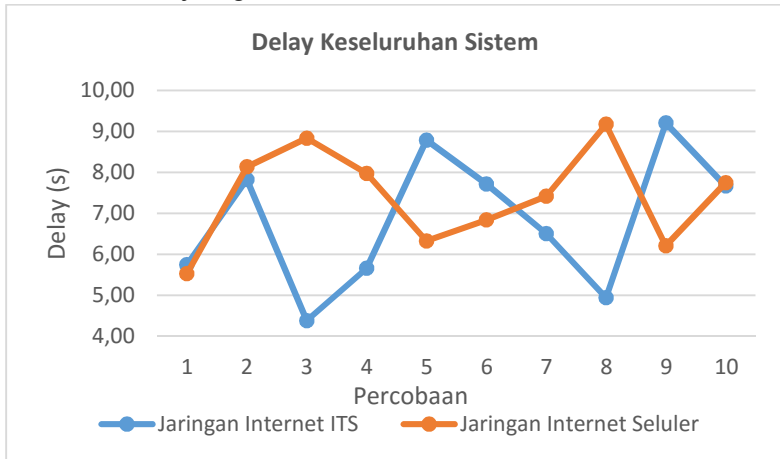
**Tabel 4.13** Hasil Pengukuran Delay Keseluruhan Sistem

Percobaan	Delay Keseluruhan Sistem (s)	
	Jaringan Internet ITS	Jaringan Internet Seluler
1	5.75	5.53
2	7.83	8.14
3	4.38	8.84
4	5.66	7.98
5	8.79	6.33
6	7.72	6.84
7	6.51	7.42
8	4.94	9.18
9	9.21	6.21
10	7.67	7.75
Rata-Rata	6.85	7.42

Seperti dijelaskan pada bab 3.5.11 bahwa pengukuran dilakukan secara manual dengan menggunakan stopwatch karena pengukuran menggunakan program timer tidak memungkinkan. Tidak memungkinkan dalam artian karena pengiriman data tidak secara langsung ke tujuan melainkan data dikirimkan pada server baru kemudian data yang ada di server dibaca oleh node gateway dan diteruskan ke node sensor.

Dari Tabel 4.13 data hasil pengukuran saat node gateway menggunakan koneksi jaringan internet kampus ITS menunjukkan rata-

rata delay sistem sekitar 6.85 detik, sedangkan saat menggunakan koneksi dari jaringan internet seluler, delay rata-rata mencapai angka 7.42 detik. Grafik pada Gambar 4.9 memperlihatkan perbedaan performansi sistem saat menggunakan jaringan internet kampus ITS dan saat menggunakan koneksi internet jaringan seluler.



**Gambar 4.9** Grafik Pengukuran Delay Keseluruhan Sistem

## LAMPIRAN

### A. Lembar Pengesahan Proposal

Jurusan Teknik Elektro – FTI

Institut Teknologi Sepuluh Nopember

TE 141599 TUGAS AKHIR – 4 SKS

Nama Mahasiswa : Tri Suliswanto  
Nomor Pokok : 2214 105 019  
Bidang Studi : Telekomunikasi Multimedia  
Tugas Diberikan : Semester Genap 2015/ 2016  
Dosen Pembimbing : 1. Dr. Ir. Wirawan, DEA.  
2. Dr. Istas Pratomo, ST. MT.

10 FEB 2016

Judul Tugas Akhir : **Rancang Bangun Sistem Kantor Pintar Berbasis Internet of Things**  
(Design and Implementation Smart Office System based on Internet of Things)

#### Uraian Tugas Akhir :

Beberapa tahun terakhir popularitas kantor pintar meningkat sangat pesat, karena keserhanaan dan kemudahan akses melalui perangkat *smartphone* dan tablet. Konsep "*Internet of Things*" erat hubungannya dalam popularitas kantor pintar. *Internet of Things*, atau dikenal juga dengan singkatan IoT, merupakan sebuah konsep yang bertujuan untuk memperluas manfaat dari konektivitas internet yang tersambung secara terus-menerus.

Sistem kantor pintar mengintegrasikan perangkat listrik maupun elektronik di sebuah kantor antara satu sama lain. Dengan melakukan integrasi ini memungkinkan kita dapat melakukan pengendalian perangkat, seperti pengaturan pencahayaan kantor, penyiraman tanaman, pengaturan suhu ruangan. Semua bisa dilakukan melalui *smartphone* anda. Dengan bantuan konsep *Internet of Things* memungkinkan akses untuk pengaturan tersebut dilakukan dimanapun dan kapanpun.

Sistem ini bekerja dengan cara melakukan pembacaan data dari *node sensor* kemudian data tersebut dikirimkan ke server yang terhubung dengan internet. Aplikasi pengguna yang terdapat di perangkat *gadget* mengambil data dari server dan melakukan visualisasi, dari data hasil visualisasi, pengguna dapat menentukan aksi apa yg harus dilakukan, penentuan aksi ini bisa dilakukan secara manual ataupun otomatis sesuai pengaturan yang telah dilakukan sebelumnya.

Pada tugas akhir ini akan dilakukan pembuatan hardware sensor node dan sistem yang akan mengintegrasikan beberapa sensor node serta desain jaringan yang akan mendukung komunikasi antar sensor node, selain itu juga akan dibuat antarmuka untuk visualisasi data. Kata Kunci : *IoT, internet of things, smart office, smart home*.

Dosen Pembimbing 1,

Dr. Ir. Wirawan, DEA.  
NIP. 196311091989031011

Mengetahui,  
Jurusan Teknik Elektro FTI – ITS  
Ketua,

Dr. Eng. Ardianto Priyadi, ST., M.Eng.  
NIP. 197309271998031004

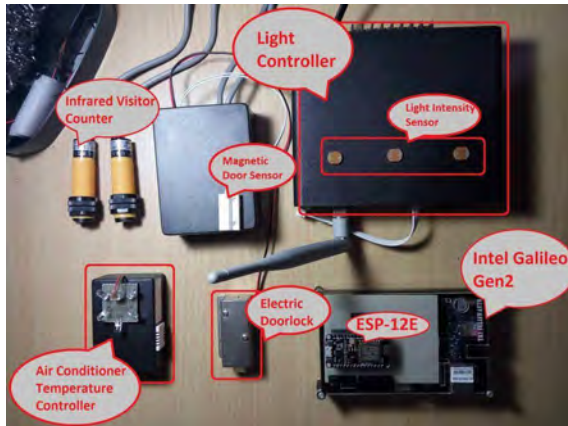
Dosen Pembimbing 2,

Dr. Istas Pratomo, ST. MT.  
NIP. 197903252003121001

Menyetujui,  
Bidang Studi Telekomunikasi Multimedia  
Koordinator,

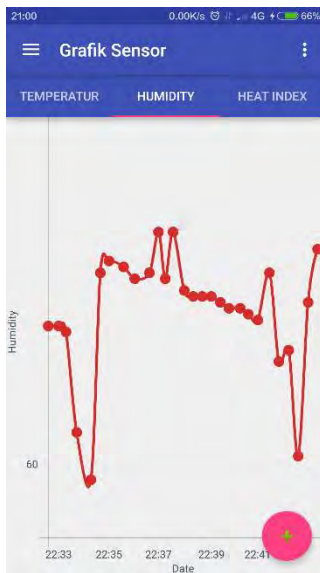
Dr. Ir. Endroyono, DEA  
NIP. 196504041991021001

## B. Foto Node



## C. Tampilan Keseluruhan Aplikasi Android





**Pintu & Sensor**

Kondisi Sensor Pintu: OK

Status Pintu: PINTU TERTUTUP

Jumlah Orang di Ruangan: 1 Orang

**Kontrol Kunci:**

☐ Kunci Pintu ☒ Buka Kunci

APPLY

**Pencahaya Ruangan**

Kontrol Mode:

☒ Otomatis ☐ Manual

Pencahaya Ruangan: OK

Sensor data: 103.42 Lux Analog data: 608

Set Threshold Lampu (Analog):

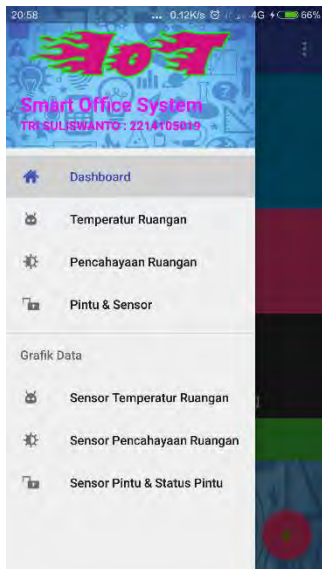
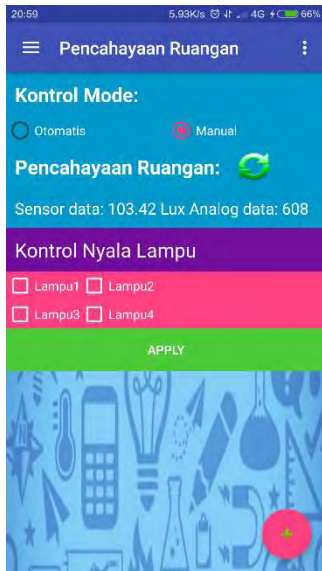
Lampu: 0-1023

Lampu: 0-1023

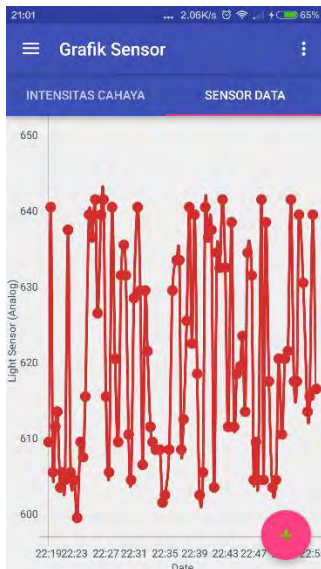
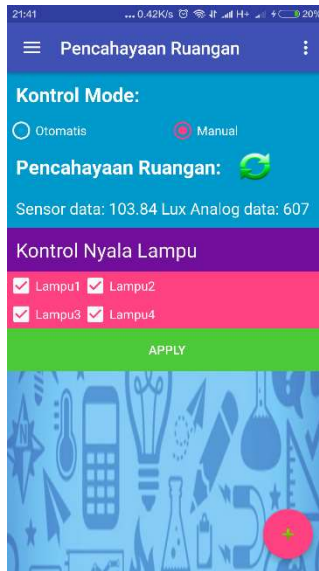
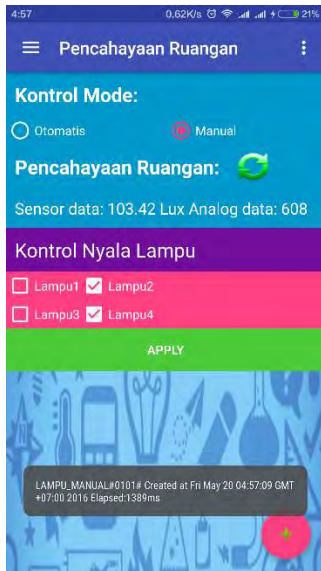
Lampu: 0-1023

Lampu: 0-1023

APPLY









#### D. Instalasi Node



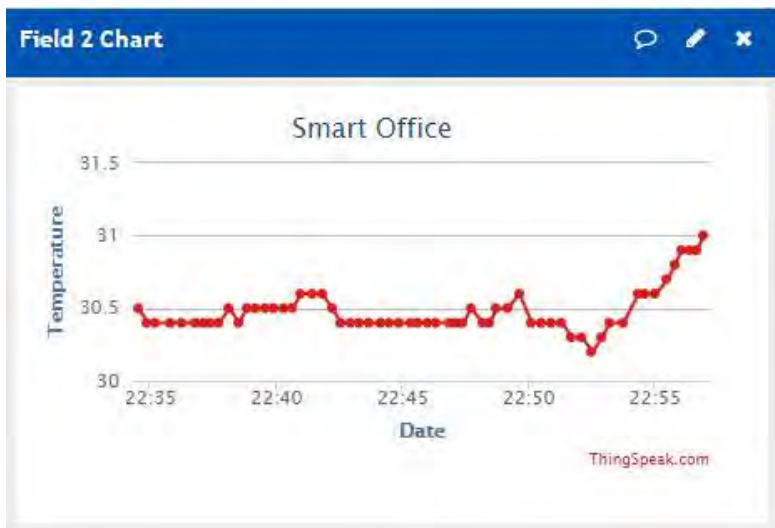


Pemasangan Node Kontroller Pencahayaan Ruangan



Pemasangan Node Kontroller Temperatur Ruangan

## E. Tampilah Grafik ThingSpeak



Field 3 Chart



### Smart Office

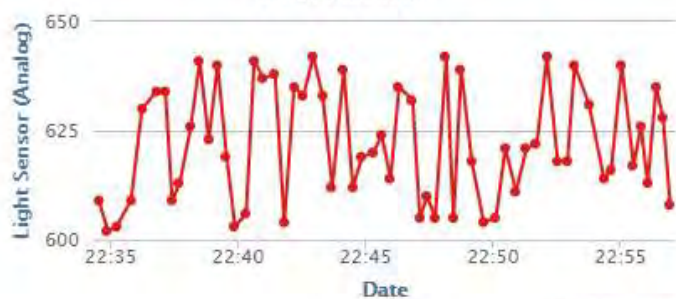


ThingSpeak.com

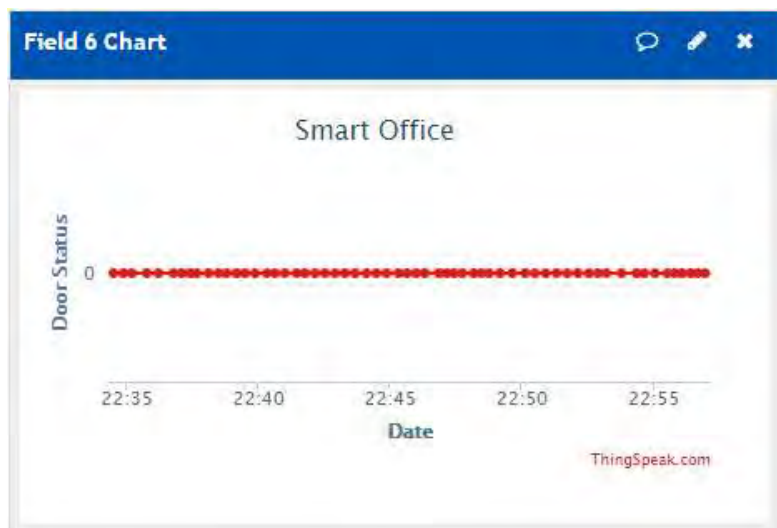
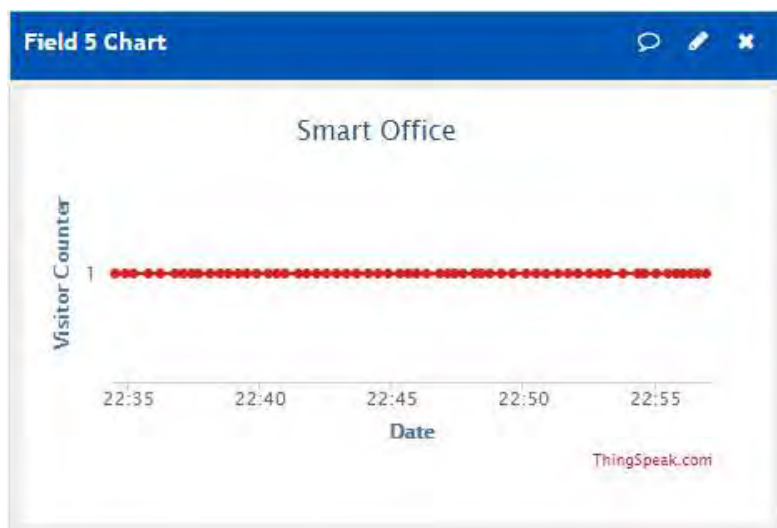
Field 4 Chart



### Smart Office



ThingSpeak.com





## F. Source Code Program

Karena source code program terlalu panjang jika dilampirkan dalam buku TA ini maka source code program saya unggah di internet, berikut adalah tautan untuk mengakses source code program:

URL: <http://bit.ly/IoT-Project>

## **BAB 5**

### **KESIMPULAN DAN SARAN**

Setelah serangkaian pengujian dan perancangan yang telah dilakukan dianalisis maka akan dapat ditarik kesimpulan. Pembahasan dari bab-bab sebelumnya pada tugas akhir ini akan menjadi bahan pertimbangan atau referensi dalam melakukan penelitian pengembangan dari tugas akhir ini.

#### **5.1 Kesimpulan**

Berdasarkan hasil perencanaan dan hasil pengujian sistem kantor pintar yang dilakukan di Laboratorium Telekomunikasi Multimedia Jurusan T. Elektro FTI di Kampus ITS Sukolilo Surabaya dapat ditarik kesimpulan.

1. Tampilan kondisi temperatur pada layar remote AC tidak merepresentasikan kondisi suhu ruangan melainkan hanya sebagai penunjuk level, Serta akurasi sensor yang digunakan pada node memiliki akurasi sangat tinggi dengan error maksimal hanya sebesar 2,10%.
2. Intensitas cahaya akan semakin kecil ketika jarak antar sumber cahaya dengan sensor semakin jauh. Node menggunakan 3 buah sensor cahaya dengan tujuan untuk meningkatkan akurasi dari pembacaan sensor.
3. Sensor penghitung jumlah orang di dalam ruangan memiliki akurasi 100% ketika sensor diletakkan di depan pintu dengan ketinggian sekitar 1 meter dari lantai, dengan catatan tidak ada pengujian yang mensimulasikan anak kecil yang masuk ke dalam ruangan.
4. Kontroller temperatur pada mode otomatis membutuhkan waktu yang lebih lama dibandingkan dengan saat tidak menggunakan sistem kantor pintar karena penggunaan sistem kantor pintar memanfaatkan jaringan internet sehingga terdapat delay. Sedangkan kontrol mode manual hanya bisa mengatur level suhu dan bukan suhu *real* ruangan.
5. Pembatasan perintah ke sistem tidak akan mempengaruhi kinerja, karena sistem menggunakan sistem antrian fifo sehingga semua perintah pasti akan dijalankan.

6. Kontroller lampu dalam mode otomatis mendukung threshold dengan resolusi 10 bit (0-1023) dengan akurasi keberhasilan menjalankan perintah hingga 100%. Mode manual memungkinkan langsung memilih lampu yang ingin dikontrol.
7. Delay pengiriman data antar node dipengaruhi oleh penempatan node di dalam ruangan, serta jarak antar node dengan node lainnya.
8. Delay pengiriman data sensor ke server bersifat tidak pasti, karena sangat tergantung pada kondisi jaringan saat dilakukan pengukuran.
9. Pengiriman data berupa perintah dari android ke server memakan waktu rata-rata sekitar 1288 ms yang berarti pengiriman data perintah tergolong cepat dan sekali lagi delay ini sangat bergantung pada kondisi jaringan saat dilakukan pengukuran.
10. Delay pengambilan data server dari android memakan waktu rata-rata sekitar 1265 ms. Ini tidak jauh berbeda dari proses pengiriman perintah ke server.
11. Delay keseluruhan sistem tidak memungkinkan dihitung oleh countrr ataupun berpatokan pada jam, karena pengiriman data tidak dilakukan secara langsung *end to end*, dan node sensor tidak memiliki *Real Time Clock (RTC)* yang dapat disinkronkan satu sama lain.
12. Untuk dapat memperoleh atau menulis data dari atau ke server maka aplikasi android dan node harus melakukan permintaan data terlebih dahulu dengan menyertakan *API KEY* dan *API ID* sebagai akun.

## 5.2 Saran

Tugas akhir ini melakukan perancangan dan implementasi sistem kantor pintar berbasis *Internet Of Things* secara umum, sehingga belum membahas secara detail mengenai sistem keamanan data dan efisiensi sumber daya yang digunakan. Beberapa hal yang bisa ditambahkan dan dikembangkan dari tugas akhir ini adalah:

1. Sistem keamanan komunikasi data antar node. Dalam tugas akhir ini masih belum menerapkan sistem keamanan sehingga tingkat keamanan masih sangat lemah karena hanya menggunakan filter alamat IP.



2. Manajemen penggunaan daya. Dalam tugas akhir ini belum terdapat sistem untuk melakukan pengendalian pemakaian daya, sehingga pemakaian daya masih belum dapat di monitor dan dikontrol.
3. Database protokol untuk jenis dan merek AC lain. Dalam tugas akhir ini hanya menggunakan 1 jenis AC merek LG sebagai studi kasus.
4. Pemisahan data masuk dan keluar dalam sensor penghitung jumlah orang di dalam ruangan agar bisa dianalisis lebih lanjut lagi.
5. Penambahan korelasi jumlah orang di dalam ruangan dengan temperatur ideal sesuai jumlah orang di dalam ruangan. Jumlah pengunjung ruangan telah berhasil dihitung namun belum terdapat korelasi antara jumlah orang di dalam ruangan dengan kondisi temperatur.

## DAFTAR PUSTAKA

- [1] Suhono Harso Supangkat, "Kembangkan Smart City sebagai Solusi Cerdas Masalah Kota di Indonesia", ICISS 2014.
- [2] Ahmed I. Abdul-Rahman, Corey A. Graves, Ph.D., "Internet of Things Application using Tethered MSP430 to Thingspeak Cloud", IEEE Symposium on Service-Oriented System Engineering, 2016.
- [3] Mathworks.Inc (2016) Getting started with Thingspeak. Available <http://www.mathworks.com/help/thingspeak/getting-started-withthingspeak.html>
- [4] Kortuem, Gerd, et al, "Smart objects as building blocks for the internet of things", Internet Computing, IEEE 14.1 (2010): 44-51
- [5] Yu Lin; Dig, D. "Refactorings for Android Asynchronous Programming", *Automated Software Engineering (ASE), 2015 30th IEEE/ACM International Conference on*, On page(s): 836 – 841
- [6] Anup W. Burange and Harshal D. Misalkar: Review of Internet of Things in Development of Smart Cities with Data Management & Privacy, *International Conference on Advances in Computer Engineering and Application (ICACEA) IMS Engineering College Ghaziabad, India, 2015*
- [7] YAN Wenbo; WANG Quanyu; GAO Zhenwei: Smart Home Implementation Based on Internet and WiFi Technology, *Proceedings of the 34th Chinese Control Conference*, Hangzhou, China, July 28-30, 2015
- [8] Para kontributor Wikipedia, "Internet of Things," Wikipedia, Ensiklopedia Bebas, [https://id.wikipedia.org/w/index.php?title=Internet\\_of\\_Things&oldid=10372248](https://id.wikipedia.org/w/index.php?title=Internet_of_Things&oldid=10372248) (diakses pada Maret 21, 2016).
- [9] Github (2016) Android Thingspeak library, available <https://github.com/MacroYau/ThingSpeakAndroid>
- [10] "HTTP 1.1 Section 5.1.1". Tools.ietf.org. Diakses tanggal 2016-02-11.

## RIWAYAT HIDUP



**Tri Suliswanto** adalah anak ketiga dari tiga bersaudara dari pasangan Zakariya dan Sulastri yang lahir di Tuban pada tanggal 13 Juli 1993. Penulis mengenyam pendidikan di SDN Gedongombo IV, Kab. Tuban, SMPN 3 Tuban, Kab. Tuban. Setelah lulus dari jenjang sekolah menengah pertama, dilanjutkan ke SMKN 1 Tuban, Kab. Tuban. Pada tahun 2011 penulis resmi menjadi mahasiswa di Politeknik Elektronika Negeri Surabaya, Surabaya pada program studi D3 Teknik Telekomunikasi dan dilanjutkan ke program Lintas Jalur S1 Teknik

Elektro Institut Teknologi Sepuluh Nopember Surabaya tahun 2014.

Penulis mengambil Tugas Akhir pada semester 4 Lintas Jalur mengenai Sistem *Smart Office* berbasis *Internet of Things* dengan dosen pembimbing Dr. Ir. Wirawan, DEA. dan Dr. Istas Pratomo, ST., MT.

Alamat email: [3sulis12@gmail.com](mailto:3sulis12@gmail.com)